

LIGHT CURVE SIMULATION AND SHAPE INVERSION FOR 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. Gregory Blaisdell

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Carolin Frueh, for her guidance and mentorship. Her feedback has guided me through undergraduate research, publishing a conference paper, applying for fellowships, and now writing this thesis. For that, and her continued support through my PhD, I am forever grateful.

I must thank my committee, Dr. Oguri and Dr. LeGrand, for their feedback and attendance at my defense. Their comments have made this thesis what it is today.

Finally, my friends and family have been an amazing resource throughout my research. They've helped me workshop ideas, find bugs in code, and proofread proposals. Most importantly, they've always been willing to look at my unusual plots.

This work was supported by Katalyst Space Technologies grant number FA6451-22-P-0019, Boeing work number SSOW-BRT-Z0722-5045, and the National Defense Science and Engineering Graduate Fellowship through the Air Force Office of Scientific Research under grant number FA9550-18-1-0154.

TABLE OF CONTENTS

LIST OF TABLES	9
LIST OF FIGURES	10
LIST OF SYMBOLS	13
ABSTRACT	16
1 Introduction	17
1.1 State of the Art	19
1.2 Research Questions and Contributions	23
2 Light Curve Simulation	25
2.0.1 The Light Curve	26
2.1 Simulating Space Object Dynamics and Properties	27
2.1.1 Orbit Propagation	27
2.1.2 Attitude Propagation	28
2.1.2.1 The Direction Cosine Matrix	28
2.1.2.2 Principal Rotation Parameters	29
2.1.2.3 Quaternions	30
2.1.3 Attitude Kinematics	31
2.1.4 Attitude Dynamics	31
2.1.5 Discrete Shape Representations	32
2.1.5.1 The Object File Format	32
2.1.5.2 Properties of Triangulated Meshes	33
2.1.6 The Bidirectional Reflectance Distribution Function	33
2.1.6.1 Lambertian	35
2.1.6.2 Phong	35
2.1.6.3 Blinn-Phong	36
2.1.6.4 Glossy	36

2.1.6.5	Cook-Torrance	37
2.1.6.6	Oren-Nayar	38
2.1.6.7	Ashikhmin-Shirley	38
2.1.7	BRDF Summary	39
2.1.8	Real Material Properties	40
2.1.9	Brightness Units	40
2.1.9.1	Irradiance	40
2.1.9.2	Apparent Magnitude	41
2.1.9.3	Normalized Irradiance	42
2.1.10	Simulating Normalized Light Curves of Convex Objects	42
2.1.11	Simulating Normalized Light Curves of Nonconvex Objects	44
2.1.11.1	Camera Projections and Terminology	44
2.1.11.2	Transformations	45
2.1.11.3	The Importance of Self-Shadowing	48
2.1.11.4	Shadow Mapping	49
2.2	Simulating The Observations	51
2.3	Time Systems	51
2.3.1	Time Scales	51
2.3.2	Julian Date	52
2.3.3	Solar and Sidereal Time	54
2.4	Coordinate Systems	54
2.4.1	Latitude, Longitude and Altitude	55
2.4.1.1	Ellipsoid	55
2.4.1.2	Geoid	56
2.4.1.3	Terrain	56
2.4.1.4	Altitude Conversions	57
2.4.2	International Terrestrial Reference Frame (ITRF)	57
2.4.3	Topocentric East North Up (ENU) Reference Frame	59
2.4.4	International Celestial Reference Frame (ICRF/J2000)	59
2.4.5	Right Ascension and Declination	61

2.4.6	Azimuth and Elevation	61
2.4.7	Coordinate Transformations Summary	62
2.4.8	Planetary Ephemerides	62
2.5	The Charge Coupled Device (CCD)	63
2.5.1	Diffraction	63
2.5.1.1	Rayleigh Criterion	64
2.5.1.2	The Airy Disk	64
2.5.2	Signal-to-Noise Ratio (SNR)	66
2.5.3	Telescope Properties	68
2.5.4	Brightness Units	68
2.5.4.1	S_{10} Surface Brightness	68
2.5.4.2	Magnitude per Square Arcsecond	69
2.5.4.3	Candela	70
2.5.4.4	Photoelectron Counts	71
2.5.5	Background Signal Sources	72
2.5.5.1	Background Source Importance	72
2.5.5.2	Astronomical Spectra	72
2.5.5.3	Airglow	74
2.5.5.4	Light Pollution	75
2.5.5.5	Twilight	77
2.5.5.6	Integrated Starlight	78
2.5.5.7	Scattered Moonlight	81
2.5.5.8	Zodiacal Light	83
2.5.5.9	Background Sampling	84
2.5.6	Sensor Effects	86
2.5.6.1	Dark Noise	86
2.5.6.2	Readout Noise	86
2.5.6.3	Truncation Noise	86
2.5.7	Observer Constraints	86
2.5.7.1	Earth Penumbra	87

2.5.7.2	SNR	89
2.5.7.3	Moon Exclusion Angle	89
2.5.7.4	Minimum Elevation	90
2.5.8	Mean Irradiance at The Observer	90
2.5.9	Sampling Noisy Light Curves	91
2.5.10	Light Curve Simulation Results	92
2.5.10.1	Case 1: Normalized Light Curve for a Cube	92
2.5.10.2	Realistic Light Curve for a Box-Wing Satellite	94
2.6	Light Curve Shape Inversion	96
2.6.1	Direct Convex Shape Inversion	96
2.6.1.1	The Extended Gaussian Image	96
2.6.1.2	EGI Optimization	97
2.6.1.3	EGI Optimization Observability	99
2.6.1.4	EGI Resampling	101
2.6.1.5	EGI Merging	102
2.6.1.6	Geometry Recovery from the EGI	103
2.6.1.7	Support Vector Optimization	105
2.6.2	Direct Nonconvex Shape Inversion	106
2.6.2.1	Nonconvex Feature Detection and Location	106
2.6.2.2	Concavity Creation	107
2.6.2.3	Model Subdivision	107
2.6.2.4	Vertex Classification	108
2.6.2.5	Vertex Displacement	110
2.6.2.6	Internal Angle Iteration	110
2.6.3	Shape Inversion With Noisy Measurements	111
2.6.3.1	Weighted Shape Interpolation	112
2.6.3.2	Shape Estimate Uncertainty	115
2.6.3.3	Spherical Nearest Neighbor Weights	118
2.6.3.4	Full Noisy Inversion Procedure	119

3	Simulated Light Curve Results	120
3.1	Accurate Satellite Models	120
3.2	Noiseless Light Curves	120
3.3	Realistic Light Curves	121
4	Shape Inversion Results	122
4.1	Convex Shape Inversion Without Noise	123
4.1.1	Object Reconstruction Improvements	123
4.2	Nonconvex Shape Inversion Without Noise	125
4.3	Convex Shape Inversion With Noise	127
4.4	Nonconvex Shape Inversion With Noise	128
4.5	Multiple Hypothesis Inversion	128
5	Recommendations	130
6	Future Work	131
7	Appendices	132
7.1	Shadow Mapping	132
7.1.1	Shading	132
7.2	Astronomical Spectra Data	133
7.2.0.1	Atmospheric Extinction	133
7.2.1	Lunar Phase Factor	133
7.2.2	Scattered Moonlight	134
7.2.3	Zodiacal Light	135
7.3	Telescope Parameters	136
7.3.0.1	Purdue Optical Ground Station	136
7.4	Wavefront OBJ Example	137
	REFERENCES	138

LIST OF TABLES

2.1	Reflection properties of real materials. MLI and white paint parameters are estimated qualitatively by the author. Significant variance may exist between objects depending on surface finishes [28].	40
2.2	WGS-84 ellipsoid model of the Earth [3].	55
2.3	Background signal importance, values generated by the author using models by Krag and Daniels [9], [67]	73
2.4	Object and station simulation parameters	93
7.1	Purdue Optical Ground Station telescope parameters	136

LIST OF FIGURES

1.1	Public tracked object catalog within $5 \cdot 10^4$ kilometers of Earth in 2000 and 2023	18
2.1	Synthetic light curves in counts, irradiance, normalized irradiance, and apparent magnitude	26
2.2	Implemented BRDFs rendered with arbitrary parameters, demonstrating the qualitative differences between lighting models	39
2.3	Total solar irradiance variations at Earth and 1 AU. The solar cycle is apparent when distance is held constant at 1 AU, yielding small variations in the peak intensity at Earth.	41
2.4	Perspective and orthographic projections with relevant camera frustum attributes labeled	45
2.5	Hubble Space Telescope ray traced shadow categorization and shading. Models from [59]	48
2.6	Normalized irradiance errors introduced by neglecting shadows for Bennu and HST. Models from [59]	49
2.7	Hubble Space Telescope shadow mapping with self (red) and horizon (blue) shadows rendered. Models from [59]	51
2.8	Time scales relative to TAI	53
2.9	EGM-96 geoid heights above the WGS-84 ellipsoid	56
2.10	MSL elevations surrounding the Purdue Optical Ground Station	57
2.11	ENU reference frame orientation at Purdue Optical Ground Station	59
2.12	Airy disk diffraction pattern	65
2.13	Signal-to-noise ratio reduction on a synthetic CCD pixel grid	67
2.14	Luminous efficiency function from [66]	70
2.15	Astronomical Spectra, atmospheric transmission and zero magnitude stellar spectrum from [9]	74
2.16	Airmass function comparison. The Van-Rhijn factor diverges to $+\infty$ while Pickering's function reaches the correct maximum of $AM(\theta_z) \approx 40$.	75
2.17	Mean airglow signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W	76
2.18	Zenith light pollution in the eastern USA, data from [69]	77
2.19	Mean light pollution signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W	78

2.20	Twilight model surface brightness at zenith as a function of solar zenith angle	79
2.21	Mean twilight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W	80
2.22	Raw image of three GEO objects with stars streaking through the background. As expected the star signals have a variety of signal-to-noise ratios. Taken by the Purdue Optical Ground station at 32.900° N, -105.533° W by Nathan Houtz.	81
2.23	Integrated starlight brightness map	82
2.24	Integrated starlight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W	83
2.25	Mean scattered moonlight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W	84
2.26	Mean zodiacal light signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W	85
2.27	Angle and area definitions for penumbra shadowing, Figure 5.20 in [9]	88
2.28	Shadow fraction in J2000, effect exaggerated by scaling the Sun radius by a factor 10	89
2.29	Attitude time history for case 1	94
2.30	Normalized light curve for case 1	95
2.31	Simplified convex, nonconvex, and open EGI nonuniqueness. Larger circles indicate greater relative areas assigned to a given normal vector.	98
2.32	Initial EGI optimization example for a cube, using 500 candidate normal vectors, the Phong BRDF with $C_d = 0.5$, $C_s = 0.5$, $n = 10$. Relative face area denoted by higher sphere opacity, reference shape shown in grey.	100
2.33	Resampled EGI for the cube data using $\phi = \frac{\pi}{20}$, 100 candidate vectors per resampled cone. Relative face area denoted by higher sphere opacity, reference shape shown in grey.	102
2.34	Merged EGI for the cube data using $\alpha = \frac{\pi}{10}$. Relative face area denoted by higher sphere opacity, reference shape shown in grey.	103
2.35	Subdivided object (black) with reference (red) with various levels of subdivision	108
2.36	Root-adjacent and free vertices	109
2.37	EGI error relationship to internal angle for the collapsed cylinder and collapsed house from [85]	110
2.38	SDF slice of the Stanford bunny model	113
2.39	SDF interpolation between a torus and an icosahedron	115

2.40	Quantity (left), quality (middle), and overall face uncertainty (right) weights . . .	117
2.41	Nearest neighbor interpolation between reference poitns (red) with uniformly distributed weights on $[0, 1]$	118
3.1	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.	120
4.1	Light curves for a 1-meter diffuse cube with $C_d = 0.5$ observed from the Purdue Optical Ground Station at 32.900° N, -105.533° W . Object is in the orbit of GOES 15 under torque-free rigid body motion with an inertia tensor $J = \text{diag}(1.0, 2.0, 3.0)$ [$kg \cdot m^2$], $q_0 = [0, 0, 0, 1]^T$, $\omega_0 = [0.01, 0.02, 0.01]$ [rad/s] . . .	122
4.2	Convex shape inversions of a cube from noisy measurements	122
4.3	Shape inversions from the initial, resampled, and merged EGIs	123
4.4	Shape inversion results for convex objects without noise in the light curve	124
4.5	Shape inversion results for nonconvex objects without noise in the light curve . .	126
4.6	Shape inversion results for convex objects with noise in the light curve	127
4.7	Shape inversion results for nonconvex objects with noise in the light curve . . .	128
4.8	Multiple hypothesis candidates from 16 nights of noisy observations of a 1-meter cylinder in the orbit of GOES 15. The object is simulated with inverted with a Phong BRDF using $C_d = 0.5$, $C_s = 0.5$, $n = 10$. Nightly light curves are each 3 hours long with data collected every 10 seconds, beginning January 1, 2023, 05:00 UTC.	129

LIST OF SYMBOLS

I	irradiance in $\left[\frac{W}{m^2}\right]$
\hat{I}	normalized irradiance in $[W]$
I_0	Irradiance of Vega $\left[\frac{W}{m^2}\right]$
m	Apparent magnitude [<i>nondim</i>]
JD	Julian date
T	Julian centuries
θ_{GMST}	Greenwich mean sidereal time
θ_r	Angular offset of the first zero of the Airy disk diffraction pattern
$C_{\text{Airy}}(\theta)$	CCD signal amplitude due to an Airy disk diffraction pattern [ADU]
k	Wavenumber
r_d	Telescope aperture radius [m]
r_o	Telescope central obstruction radius [m]
d	Telescope aperture diameter [m]
A_{aperture}	Telescope aperture area [m^2]
A_{eff}	Telescope effective aperture area [m^2]
f	Telescope focal length [m]
λ	Wavelength [m]
$FWHM$	Full width at half maximum
$C_{\text{Gauss}}(\theta)$	CCD signal amplitude due to a Gaussian approximation of the Airy disk [ADU]
STRINT(λ)	Representative zero apparent magnitude star irradiance spectrum $\left[\frac{W}{m^2 \cdot m}\right]$
QE(λ)	Quantum efficiency spectrum $\left[\frac{\text{ADU}}{m}\right]$
ATM(λ)	Atmospheric transmission spectrum $\left[\frac{1}{m}\right]$
$K_{cd}(\lambda)$	Luminous efficacy spectrum $\left[\frac{lm}{W}\right]$
SINT	CCD ADU conversion factor $\left[\frac{\text{ADU}}{W \cdot m^{-2} \cdot s}\right]$
SUN(λ)	Solar irradiance spectrum $\left[\frac{W}{m^2 \cdot m}\right]$
GLINT(λ)	Airglow radiance spectrum $\left[\frac{W}{m^2 \cdot m \cdot sr}\right]$
AINT	Intermediate airglow signal $\left[\frac{1}{s \cdot sr}\right]$

θ_z	Zenith angle [rad]
$\text{AM}(\theta_z)$	Relative airmass function [nondim]
s_{pix}	Telescope pixel scale $\left[\frac{\text{arcsec}}{\text{pix}}\right]$
Δt	CCD integration time [s]
$B_{poll,z}$	Zenith light pollution brightness in magnitudes per square arcsecond
$\bar{S}_{airglow}$	Mean airglow signal [ADU]
γ	Solar zenith angle [deg]
$B_{twi,z}$	Zenith twilight brightness in magnitudes per square arcsecond
$\bar{S}_{twilight}$	Mean twilight signal [ADU]
BINT	Zero magnitude starlight signal $\left[\frac{\text{ADU}}{\text{s}}\right]$
\bar{S}_{star}	Mean integrated starlight signal [ADU]
F_{rs}	Moonlight Rayleigh scattering radiance spectrum $\left[\frac{W}{m^2 \cdot m \cdot sr}\right]$
F_{ms}	Moonlight Mie scattering radiance spectrum $\left[\frac{W}{m^2 \cdot m \cdot sr}\right]$
F_{mt}	Total scattered moonlight radiance spectrum $\left[\frac{W}{m^2 \cdot m \cdot sr}\right]$
$f(\theta)$	Lunar brightness phase function [nondim]
\bar{S}_{moon}	Mean scattered moonlight signal [ADU]
\bar{S}_{zod}	Mean zodiacal light signal [ADU]
$\lambda_{background}$	Mean of background signal Poisson distribution [ADU]
\hat{n}	Face outward pointing unit normal vector
(v_1, v_2, v_3)	First, second, and third vertices $v_i \in \mathbb{R}^3$ on a given triangular face
h_i	Support of the ith face
\vec{E}	Extended Gaussian Image
f_r	Bidirectional Reflectance Distribution Function
L	Illumination direction unit vector
O	Observation direction unit vector
H	Halfway unit vector
C_d	Coefficient of diffuse reflection
C_s	Coefficient of specular reflection
C_a	Coeffieicnt of absorption

n

Specular exponent

ABSTRACT

Characterizing unknown space objects is an important component of robust space situational awareness. Estimating the shape of an object allows analysts to perform more accurate orbit propagation, probability of collision, and inference analysis about the object’s origin. Due to the sheer distance from the camera combined with diffraction and atmospheric effects, most resident space objects of interest are unresolved when observed from the ground with electro-optical sensors. State of the art techniques for object characterization often rely on light curves — the time history of the object’s observed brightness. The brightness of the object is a function of the object’s shape, material properties, and attitude profile, along with the observation geometry. The process of measuring real light curves is complex, involving the physics of the object, the sensor, and the background environment. The process of recovering shape information from brightness measurements is known as the light curve shape inversion problem. This problem is ill-posed without further assumptions: modern direct shape inversion methods require that the attitude profile and material properties of the object is known, or at least can be hypothesized. This work describes improvements to light curve simulation that faithfully model the environmental and sensor effects present in true light curves, yielding synthetic measurements with more accurate noise characteristics. Having access to more accurate light curves is important for developing and validating light curve inversion methods. This work also presents new methods for direct shape inversion for convex and nonconvex objects with realistic measurement noise. In particular, this work finds that improvements to the convex shape inversion process produce more accurate, sparser geometry in less time. The proposed nonconvex shape inversion method is effective at resolving singular large concave feature.

1. Introduction

Humankind has been creating space debris since the dawn of the space age [1]. Early missions like Vanguard 1, launched in March of 1958, set a precedent by leaving both their satellite and the launch vehicle's upper stage in orbit, both of which are still in orbit in 2023 [2]. Vanguard was launched into Low Earth Orbit (LEO) — defined by the European Space Agency (ESA) as any orbit with an altitude below 2000 kilometers [1]. Above LEO lies Medium Earth Orbit (MEO) for orbital altitudes between 2000 and 31570 kilometers, and Geostationary Earth Orbit (GEO) between 35586 and 35986 kilometers altitude [1]. Half a century of increasingly frequent launches has created a space environment cluttered with thousands of debris objects, increasing the number of serious conjunction events that may require avoidance maneuvers for large satellites in LEO to over 100 during 2021 [1]. The rapidly changes in the space environment over the last two decades is illustrated in Figure 1.1. While very few mission-ending collisions are occurring on a yearly basis in the early 2020s, simulations predict over 200 catastrophic collisions per year if the trend in new launches and disposal practices continue [1]. This uncontrolled proliferation of human-made space debris puts space operations at risk. High-profile satellite collisions like Iridium-Cosmos in 2009 have added fuel to the fire, producing shells of debris that further pollute LEO [3]. Anti-satellite tests carried out by the USA, Russia, China, and India since the 1960s see nations destroying their own satellites, projecting military strength at the cost of creating more debris [3]. Beyond LEO in Geostationary Transfer Orbit (GTO), exploding launch vehicle upper stages produce large amounts of debris [1]. While higher orbits are not yet as polluted as LEO, they do not decay due to atmospheric drag, allowing debris objects to remain in the environment indefinitely [3].

In the context of the modern space environment, determining the current state and predicting the future dynamics of space objects is critical for many areas of Space Domain Awareness (SDA) [4]. While the current orbits of objects can be determined accurately from astrometry — through passive optical imagery or active radar — their future dynamics are perturbed by non-conservative forces driven by their shape, attitude profile, and material properties that cannot be observed directly. In particular, objects in orbits with altitudes

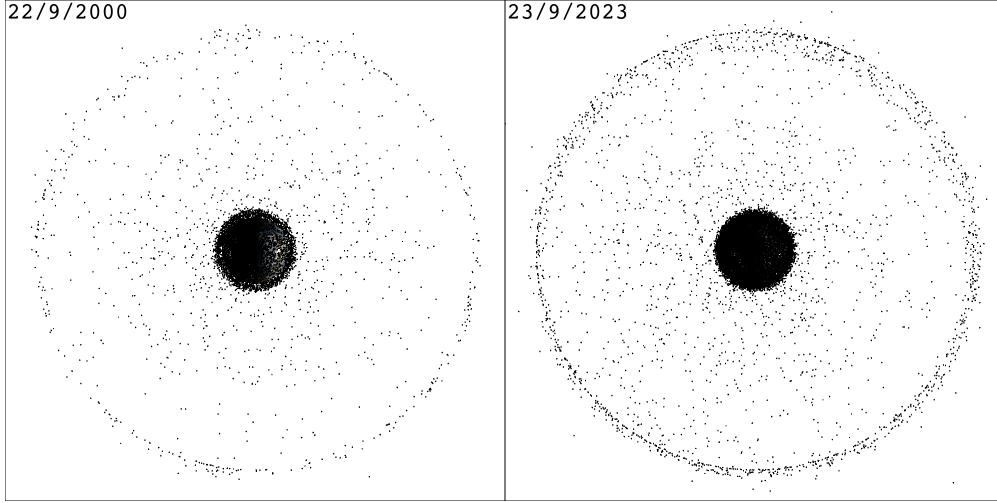


Figure 1.1. Public tracked object catalog within $5 \cdot 10^4$ kilometers of Earth in 2000 and 2023

higher than LEO are most efficiently observed with optical telescopes as the power required for radar scales with the square of the distance [4]. Because optical observations are already commonly used to characterize the orbit of the objects in orbits past LEO, it is advantageous to use the same existing sensors — or in some cases even the same images — to extract these other useful characteristics.

Characterizing an object’s shape, attitude, and material properties is fundamentally difficult as distance from the sensor and atmospheric turbulence leaves only a distribution of brightness in the image [5]. One part of the remaining information is the total brightness of the object, and this value over time is known as the light curve. Optical brightness observations are fundamentally limited by background and sensor noise [4]. Light curves are computed from observed optical data by estimating the background mean of the image, identifying which pixels of the image likely belong to the object, subtracting the mean background level from those pixels, and calibrating the remaining object signal using known stars elsewhere in the image [6]. Each image must also be monitored for contamination from background stars and over- and under-exposures [7]. Despite these realities, the light curve is a function of the parameters of interest: the object’s shape, attitude, and material properties [5], [8]. Solving light curve shape inversion in a general case would enable robust

active debris removal, anomaly detection, and collision avoidance, all of which are benefitted by accurate shape information.

Due to the environmental noise and fundamental physical limitations on the processes driving the light curve, the measured brightness is dependent on the overall brightness and hence varies from data point to data point in the light curve. Furthermore, given the Poisson nature of the light collection process, a constant Gaussian assumption of the measurement noise in the light curve may not be suitable [5], [9]. A realistic representation of the light curve can only be achieved by accounting for the physical processes simulating the lighting and measurement process, followed by the measurement reduction and correction processing steps.

1.1 State of the Art

The state of the art in light curve simulation differs between approaches and the object class under study. Kaasalainen and Torppa, as well Fan, Friedman, Kobayashi, and Frueh employ a Lambertian model for convex objects with a facetwise ray tracing scheme for nonconvex objects [5], [10]–[14]. This approach has the advantage of being simple, but can be computationally intensive for complex objects. Furfaro et al. [15] and Cabrera and Bradley [16], [17] use a simple Lambertian model with no self-shadowing. Many more authors apply a more specialized non-Lambertian Bidirectional Reflectance Distribution Function (BRDF) for their lighting [18]–[20]. Few papers in the literature focus solely on light curve simulation, most treating simulation as a background consideration for the real goal of inversion. In the minority, Allworth et al. developed a ray traced light curve simulator in based on Blender’s cycles renderer, allowing them to account for photorealistic shadowing and motion blur [21], [22]. Often, advances in light curve simulation are driven by developments in the field of computer graphics. The ray tracing schemes rely on work by Möller and Trumbore [23], while the more complex reflection functions used to simulate light transport are all contributions from the computer graphics community [24]–[28]. Throughout the literature, there is a clear gap between the simulated light curves and their observed counterparts. Due to the difference in quality, authors often treat real and simulated data very differently [22]. This

work presents a physically-based lighting, shadowing, and noise model to produce synthetic light curves of similar quality to observed data, enabling more robust validation of the presented inversion techniques.

Light curve shape inversion was first investigated by Russell in 1906, who proposed a spherical harmonic representation that could be fit to an asteroid shape [29]. Russell noted that there would be ambiguity in the shape solution such that many solutions would fit the data equally well. The next major contribution to the field was due to Kaasalainen and Torppa in 2000, who successfully reconstructed the shapes of asteriods by directly optimizing the directions and areas of candidate faces — encapsulated by the so-called Extended Gaussian Image (EGI) — to find a convex shape that produces a similar light curve [10], [30]. Once the EGI is estimated, Kaasalainen and Torppa recover the vertices and faces of the corresponding convex object using a result of Minkowski and a nonlinear, convex optimization problem implemented by Little [31], [32]. Any EGI-based method in the asteroid or human-made object shape inversion literature uses some variation of this final stage to reconstruct the final estimated geometry. Kaasalainen and Torppa also addressed nonconvex shape inversion by optimizing a spherical harmonics shape representation to reconstruct the largest nonconvex features of an asteroid, noting that smoothness regularization was sometimes needed to prevent the shape from degenerating [30]. In the work of Kaasalainen and Torppa, the EGI optimization takes place in a single step as the asteroid shapes under study do not have sparse EGIs. By contrast, this work introduces stages that increase shape accuracy and lower computation time by leveraging the natural sparsity of human-made objects. Durech and Kaasalainen extended on this work in 2003 by investigating the observability of nonconvex features in asteroid light curves, finding that concave features are often observable only at high phase angles, supporting the conclusion that robust nonconvex shape inversion requires very different considerations than its convex counterpart [33]. In 2022, Chng et al. proposed a method to determine a optimal spin pole and convex shape via the EGI, offering computational benefits over Kaasalainen and Torppa while guaranteeing global optimality in the solution with respect to the input brightness data, while being limited to convex shape estimates[34]. Using the methods originally proposed by Kaasalainen and Torppa, a collaborative effort of dozens of observatories lead to the publication of Database of Aster-

oid Models from Inversion Techniques (DAMIT), a publicly-available repository of convex asteroid models [35]. As of October 2023, DAMIT currently hosts 16,086 models for 10,751 asteroids [36].

Shape inversion for human-made space objects differs from the asteroid inversion in a few important aspects. More diverse methods exist, being generally segmented into EGI-based methods drawing from the asteroid shape inversion literature, filter-based methods for simultaneous attitude and shape solutions, and machine learning for classifying object shape from the light curve. Due to the increased number of unknowns in the material properties and attitude profile when observing an arbitrary human-made object, the inverted light curves are often simulated as part of the same work. This highlights the importance of realistic light curve simulation to effectively test proposed inversion methods.

Direct shape inversion for human-made space objects was first investigated by Calef et al. who adopted Kaasalainen and Torppa’s methods applied to multispectrum measurements to reduce the ambiguities of the different material properties common in human-made objects [37]. Bradley and Axelrad also used asteroid inversion techniques to recover convex approximations of CubeSats, rocket bodies, and box-wing satellites using the inversion codes developed and released by Kaasalainen, yielding good results for rocket body-like shapes but limited success for box-wing satellites and other high area-to-mass ratio (HAMR) objects [17]. The most recent major contributions to the direct shape inversion literature are due to Fan and Frueh, who inverted the shape of convex human-made objects from noisy light curves using the EGI with a multi-hypothesis scheme to reduce the ambiguity introduced by noisy measurements [5], [38], [39]. Fan notes that full observability is crucial for successful direct shape inversion, pointing to work by Friedman and Frueh, who quantified the observability of EGI inversion to inform sensor tasking schemes [12], [40]. Cabrera et al. applied area regularization to Fan and Friedman’s methods, achieving more accurate convex shape estimates and finding that natural constraints on the EGI area optimization renders the problem estimatable before it becomes classically observable [16].

Throughout the shape inversion literature, two themes are clear. Effective and efficient methods for nonconvex shape inversion for human-made objects are needed, and existing convex inversion methods have not been designed to work with realistic measurement noise.

This work seeks to address both of these challenges by presenting a method for inverting large singular concave features in addition to a scheme for robustly inverting convex and nonconvex shapes with physically-based measurement noise.

Outside of the asteroid-inspired EGI methods, the literature falls into two broad categories: filter-based inversion and machine learning categorization. Each offers different advantages while imposing unique limitations. Filter-based shape inversion was been pioneered by Linares through work with various co-authors. These filter-based methods often seek to perform multiple types of object characterization simultaneously, estimating attitude and material properties in addition to shape [18], [41], [42]. Because the input data for filter-based approaches is still only unresolved brightness measurements, estimating more properties in an already ill-posed problem requires a loss of fidelity in the solution elsewhere. Often, the shape model is highly simplified to make the problem more tractable [18], [41], [42]. Linares et al. have implemented unscented Kalman filters [41], multiple-model adaptive estimation algorithms [42], and adaptive Hamiltonian Markov chain Monte Carlo schemes [18] which achieve good results for simple shapes, but have not been tested on complex and realistic geometries [18]. In general, filter-based approaches are limited by the nonlinearity of highly specular and complex human-made objects, but require less information to run. Direct inversion methods that use the EGI require more *a priori* information, but are able to deliver more accurate shape estimates.

By contrast, machine learning categorization methods indirectly recover shape information by predicting which class of objects an observed light curve belongs to. Linares and Furfaro used a deep convolutional neural network to classify novel light curves as rocket bodies, payloads, or debris, achieving good classification accuracy at the cost of uncertainty about how the model would behave for light curves collected for objects outside of its training dataset [43]. Other authors, including Kerr et al. and McNally et al. have adapted the architecture developed by Furfaro et al. to classify novel light curves into an extended set of object types, demonstrating that these models are flexible enough to differentiate between many object types and attitude profiles, although with higher error rates [19], [44]. Allworth et al. applied transfer learning to classify real measurements using a synthetically-trained model, supporting the applicability of these approaches to operational decision-making [22].

There has also been significant work published on extracting light curves of human-made objects from real optical observations. Schildknecht et al. used color photometry to investigate isolate material properties of high area-to-mass ratio (HAMR) objects in GEO [6]. Karpov et al. used wide-field monitoring system to collect light curves from LEO objects [45]. Benson et al. collected light curves from retired GOES, Inmarsat, and Astra satellites in geosynchronous orbit to characterize their spin states [46]. Koshkin et al. collected light curves of TOPEX/Poseidon, among other inactive satellites, to determine their spin poles and rates [47]. Wang et al. collected light curves from GOES-8, an active GEO satellite, and simulated material properties and attitude profile to attribute peaks in its observed brightness to different parts of the spacecraft [48].

1.2 Research Questions and Contributions

This work seeks to address three major gaps in the state of the art.

1. Light curve simulation methods do not model much of the physics of the observation, leading to noiseless measurements or noisy values that do not match the real-world processes.
2. Direct shape inversion for human-made objects is limited to convex objects, a problem given that most human-made space objects are highly nonconvex.
3. Noisy measurements degrade the performance of standard shape inversion procedures. Existing noisy shape inversion procedures assume convexity in the truth object and a constant noise level, two assumptions that often do not hold in reality.

These gaps lead to three research questions that guided this work.

1. What physical processes influence the mean and variance of the light curve, and how can these processes be efficiently modeled?
2. How can concave features in the object's geometry be identified from the light curve and introduced into the shape estimate?

3. What effect does noise have on the shape guesses produced by standard algorithms, and how information from multiple inaccurate shape estimates be combined into a better final estimate?

This work presents three contributions, aligned with the identified research questions.

1. A high-fidelity light curve simulator was developed, efficiently modeling self-shadowing, background environmental effects, and sensor effects to produce light curves with or without realistic noise.
2. Improvements to the convex shape inversion procedure through the addition of resampling and merging stages produce sparser geometry in less computation time. A new method for estimating large and unilateral concavities is also developed which imposes no new assumptions on the convex inversion scheme.
3. For noisy shape inversion, a novel method for estimating the uncertainty in the estimated geometry is developed, leading to a weighted averaging procedure to combine the most reliable features of multiple shape estimates.

2. Light Curve Simulation

Light curve simulation and inversion requires background knowledge in attitude kinematics, time and coordinate systems, computer graphics, and photometry. This section is broken into two major parts: simulating the object and its properties, and simulating the observations. The object simulation discussion proceeds with:

1. Orbit and attitude propagation,
2. Discrete mesh representations of shapes,
3. Reflectance functions,
4. Simulating light curves for convex objects,
5. Simulating light curves for nonconvex objects.

The discussion of the observation simulation proceeds with:

1. Time and coordinate systems,
2. Modeling the telescope sensor,
3. Modeling the background,
4. Observer constraints.

This overall ordering for the simulation section of this work was chosen to promote an understanding of the object and its reflection properties in a simplified manner before expanding on the physics of the observer to introduce noise and accurate observation intervals to the light curve. To preface this section, the light curve and its common units must first be defined.

2.0.1 The Light Curve

The light curve expresses the observed brightness of an object over time. In literature, that brightness is commonly expressed in terms of irradiance, apparent magnitude, or photoelectron counts. This work also considers the normalized irradiance for some simulation and inversion tasks. Converting between these representations is accomplished with Eqs 2.80 and 2.31.

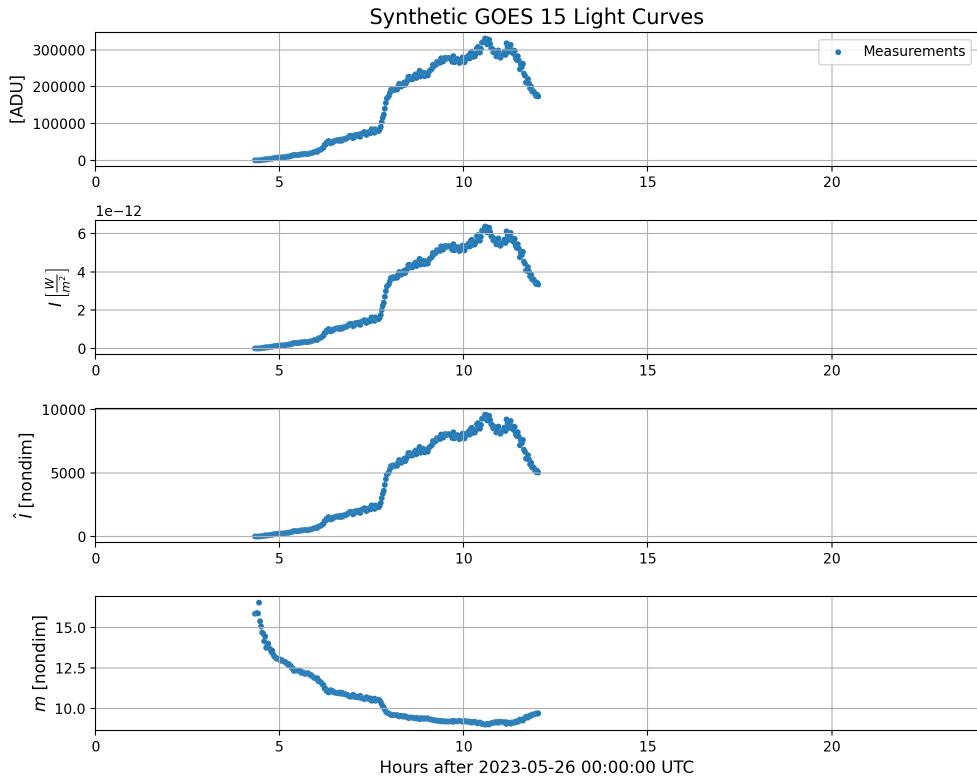


Figure 2.1. Synthetic light curves in counts, irradiance, normalized irradiance, and apparent magnitude

Figure 2.1 details this equivalence with the same light curve expressed in these four unit systems. This figure also displays an important characteristic of real measured light curves: they are always contaminated with noise. As a result, the "true mean" signal in Figure 2.1 cannot be obtained through the raw measurements, and is merely computed here as a stepping stone to the measured digital signal in the CCD. The sources of this noise are described in detail in Section 2.5.

2.1 Simulating Space Object Dynamics and Properties

2.1.1 Orbit Propagation

There are many environmental perturbations that influence the natural motion of space objects in orbits around Earth. In LEO, drag and the J_2 term of Earth's gravity field due to equatorial oblateness are the dominant perturbations [4]. In MEO and GEO, solar radiation pressure (SRP) and the third-body effects of the Sun and Moon become increasingly important while drag is negligible [4]. For the purposes of accurate light curve simulation, the orbit propagation must be accurate enough for the hour to week-long scenarios, and it should also be easy to access real initial conditions for objects of interest near the propagation dates. To access historical and predicted trajectories of real space objects in the public catalog, Two-Line Element sets (TLEs) are used. TLEs are short, standardized collections of orbital elements and mission information, produced from observations of the object in question close to the publication date [3]. TLEs are available programatically through Space-Track, a service provided by the US Space Force's 18th Space Defense Squadron [49]. The Simplified General Perturbations 4 (SGP4) propagator is used to propagate near-Earth objects from TLEs. SGP4 models the secular effects of the first few zonal terms of Earth's gravity field, drag through an exponential density atmosphere, and some of the effects of third bodies like the Sun and Moon [3]. It should be noted that an improved propagator known as SGP4-XP has been developed to better model SRP, cislunar objects, gravitational harmonics, and atmospheric drag [50]. TLEs for SGP4-XP and SGP4 are not interchangeable [50]. The SGP4 propagator takes as input a Two-Line Element (TLE) set along with the Julian dates to propagate the object to and returns the position and velocity in TEME [3]. While SGP4 does not model some important perturbations like SRP, the availability of frequently updated TLEs for regularly tracked objects alleviates most concerns, although errors on the order of a few kilometers are still expected for LEO objects [3].

Orbital propagation for the simulations performed in this work was done using SGP4 with TLE interpolation. For a given set of Julian dates and the satellite name in question, Space-Track is queried to find all TLEs produced for that object in the interval of interest. SGP4 is run for all Julian dates and all TLEs, and the final position at each Julian date is

linearly interpolated between the most recent past TLE and the next future TLE. If there is only one neighboring TLE, its position is used without interpolation.

2.1.2 Attitude Propagation

In order to effectively compute reflections and shadows on the surfaces of space objects, that object must have an assigned orientation in inertial space. While defunct satellites and space debris are slowly spun up by the Yarkovsky-O’Keefe-Radzievskii-Paddack (YORP) effect, the short-term motion of non-operational space objects is well-approximated by torque-free motion [51]. Developing the kinematics of torque-free motion first requires an understanding of common kinematic representations of orientation.

The orientation — or attitude — of a rigid body in three dimensions is implicitly understood to be relative to some other reference frame. The direction of a unit vector can be expressed with two numbers — the azimuth and elevation of that vector. Naïvely, this could be extrapolated to conclude that six numbers are needed to express an orientation. Because the basis vectors form an orthonormal set $\{\hat{b}_1, \hat{b}_2, \hat{b}_3\}$, it follows for a right-handed system that $\hat{b}_3 = \hat{b}_1 \times \hat{b}_2$, $\hat{b}_2 = \hat{b}_3 \times \hat{b}_1$, and $\hat{b}_1 = \hat{b}_2 \times \hat{b}_3$. Each of these equations constrains one further degree of freedom, revealing that a minimum of three quantities are necessary to express the relative orientation of two reference frames. This minimum bound does not make any statements about the usefulness of three element sets; at least four dimensions are needed to remove singularities [52].

2.1.2.1 The Direction Cosine Matrix

The direction cosine matrix (DCM) is a 3×3 symmetric, orthogonal matrix, expressing the three basis vectors of one frame in another. This amounts to projecting each basis vector in the initial frame onto each basis vector of the final frame; the cosine of the angle between the compared vectors. It is notated 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 a vector ${}^N\mathbf{r}$ in the \mathcal{N} frame to the \mathcal{B} frame as ${}^B\mathbf{r}$ [53]:

$${}^B\mathbf{r} = [\mathcal{BN}] {}^N\mathbf{r}. \quad (2.1)$$

The orthogonal property of the DCM implies $[\mathcal{BN}]^{-1} = [\mathcal{BN}]^T$ such that $[\mathcal{BN}]^T = [\mathcal{NB}]$ markley2016.

2.1.2.2 Principal Rotation Parameters

Another common attitude representation is the Euler angle-axis set, otherwise known as principal rotation parameters [52]. 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 $\theta \in [0, 2\pi]$ [52]. The set $\{\hat{\lambda}, \theta\}$ is known as a principal rotation parameter, abbreviated PRP hereafter. The DCM is mapped to the PRP representation with [53]:

$$\begin{aligned} \theta &= \cos^{-1} \left(\frac{1}{2} [C_{1,1} + C_{2,2} + C_{3,3} - 1] \right), \\ \hat{\lambda} &= \frac{1}{2 \sin \theta} \begin{bmatrix} C_{2,3} - C_{3,2} \\ C_{3,1} - C_{1,3} \\ C_{1,2} - C_{2,1} \end{bmatrix}, \end{aligned} \quad (2.2)$$

where $C_{i,j}$ refers to the i th row and j th column of C [53]. The mapping from PRP to DCM is also relatively straightforward [53]:

$$C = I_3 + \sin \theta [\hat{\lambda} \times] + (1 - \cos \theta) [\hat{\lambda} \times]^2, \quad (2.3)$$

where $[v \times]$ is the matrix cross product operator, defined on $\mathbf{v} \in \mathbb{R}^3$ as [53]:

$$[\mathbf{v} \times] = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \quad (2.4)$$

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 $\hat{\lambda}$.

2.1.2.3 Quaternions

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 [52]:

$$\mathbf{q} = \begin{bmatrix} \hat{\lambda} \sin(\theta) \\ \cos(\theta) \end{bmatrix}. \quad (2.5)$$

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, the vector portion of the quaternion is referenced with $\mathbf{q}_{1:3}$. The quaternion can be mapped back to the PRP [52] via:

$$\begin{aligned} \theta &= \cos^{-1}(q_4) \\ \lambda &= \frac{\mathbf{q}_{1:3}}{\sin \theta}. \end{aligned} \quad (2.6)$$

The quaternion maps to the DCM [52] via :

$$C = \begin{bmatrix} -q_2^2 - q_3^2 + q_1^2 + q_4^2 & 2q_1q_2 + 2q_3q_4 & 2q_1q_3 - 2q_2q_4 \\ 2q_1q_2 - 2q_3q_4 & -q_1^2 - q_3^2 + q_2^2 + q_4^2 & 2q_1q_4 + 2q_2q_3 \\ 2q_1q_3 + 2q_2q_4 & 2q_2q_3 - 2q_1q_4 & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} = \Xi(q)^T \Psi(q). \quad (2.7)$$

In Eq 2.7, Ψ is defined to be [52]

$$\Psi = \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}. \quad (2.8)$$

2.1.3 Attitude Kinematics

Quaternion kinematic differential equations are chosen for simulating object attitude profiles as they have no singularity and produce very smooth dynamics that are easy to integrate when compared to three-variable representations that possess singularities. Given the current orientation quaternion $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ and angular velocity $\omega = [\omega_1, \omega_2, \omega_3]^T$ the quaternion derivative is computed via [52]:

$$\begin{bmatrix} \dot{\epsilon}_1 \\ \dot{\epsilon}_2 \\ \dot{\epsilon}_3 \\ \dot{\epsilon}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \epsilon_4 & -\epsilon_3 & \epsilon_2 & \epsilon_1 \\ \epsilon_3 & \epsilon_4 & -\epsilon_1 & \epsilon_2 \\ -\epsilon_2 & \epsilon_1 & \epsilon_4 & \epsilon_3 \\ -\epsilon_1 & -\epsilon_2 & -\epsilon_3 & \epsilon_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix}. \quad (2.9)$$

2.1.4 Attitude Dynamics

Rigid body dynamics can be easily expressed in the body principal axes for an object with the diagonal inertia tensor $J = \text{diag}(J_1, J_2, J_3)$ with an arbitrary torque $\mathbf{M} = [M_1, M_2, M_3]^T$ in the same frame via [52]:

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} (M_1 + I_2\omega_2\omega_3 - I_3\omega_2\omega_3) / J_1 \\ (M_2 - I_1\omega_1\omega_3 + I_3\omega_1\omega_3) / J_2 \\ (M_3 + I_1\omega_1\omega_2 - I_2\omega_1\omega_2) / J_3 \end{bmatrix}. \quad (2.10)$$

Equations 2.10 and 2.9 are numerically integrated to yield the orientation time history which is necessary for later light curve simulations. Often, $\mathbf{M} = \mathbf{0}$ is chosen to simulate

the torque-free evolution of an initial orientation and spin state, neglecting small attitude perturbations over the simulation interval.

2.1.5 Discrete 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, a shape may be defined by a set of signed distance functions (SDFs). 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 shape. The object can then be rendered via ray marching. A ray is cast from the camera out into the scene for each pixel of the screen, each performing distance queries along its length until it intersects the object or diverges.

By contrast, an explicit shape representation creates complex 3D geometry from simple 2D building blocks. In the most common case, object faces are defined by triangles. This means that at the scale of the individual faces, 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 light curve shape inversion. Human-made space objects like most satellites are composed of flat faces, with the exception of parabolic antennas and cylindrical rocket bodies.

2.1.5.1 The Object File Format

One common text file format for 3D model files is `.obj`, developed by Wavefront Technologies in the early 1990s [54]. Each OBJ file consists of a list of vertex positions and face definitions, with optional vertex normals and tangents. An `.obj` listing for a cube is included for reference in Appendix 7.4. Given the vertex positions and adjacency information stored in the model file, useful properties of the object can be computed for use later in both light curve simulation and shape inversion.

2.1.5.2 Properties of Triangulated Meshes

For each triangular face F_i of the model defined by vertices $F_i = \{v_1, v_2, v_3\}$, the outward-pointing face normal is computed with:

$$\hat{n} = \frac{(v_2 - v_1) \times (v_3 - v_1)}{\|(v_2 - v_1) \times (v_3 - v_1)\|_2}. \quad (2.11)$$

The face area is computed with:

$$a = \frac{\|(v_2 - v_1) \times (v_3 - v_1)\|_2}{2}. \quad (2.12)$$

The support of the i th face — the perpendicular distance from the origin to the plane defining the face — is computed with the position of any vertex on that face, i.e. the first vertex $v_{i,1}$, and the face normal vector \hat{n}_i :

$$h_i = v_{i,1} \cdot \hat{n}_i. \quad (2.13)$$

The volume of the object is computed:

$$V = \frac{1}{3} \sum_{i=0}^{|F|} \vec{h}_i \cdot \vec{a}_i. \quad (2.14)$$

In Eq 2.14, $|F|$ is the number of faces defining the object. \vec{h} and \vec{a} are column vectors collecting all face supports and areas. The Extended Gaussian Image, a quantity defined in 2.6.1.1, is computed row-wise for the i th face with:

$$\vec{E}_i = \vec{a}_i \vec{n}_i. \quad (2.15)$$

2.1.6 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, all relevant optical interactions must be modeled.

In broad terms, this boils down to determining how the object is illuminated, how it casts shadows on itself, and how it is observed.

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 these microfacets. If the facet normals are distributed in biased orientations, the macroscopic surface may show anisotropy, leading to the appearance of brushed metal. If the microfacet normals are at large angles to each other, the surface may appear dull as the direction of the outgoing light may be largely 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 should the macroscopic outcomes of the microscopic interactions of incident light on a surface be modeled? The bidirectional reflectance distribution function (BRDF) is a tool from computer graphics that addresses this 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 [55]:

$$f_r(\mathbf{x}, L, O) = \frac{d\mathcal{R}(\mathbf{x} \rightarrow O)}{d\mathcal{I}(L, \mathbf{x})}. \quad (2.16)$$

In Eq 2.16, $\mathbf{x} \in \mathbb{R}^3$ is the point on the object's surface where the BRDF is evaluated. $L \in \mathbb{S}^2$ is the incoming illumination unit vector and $O \in \mathbb{S}^2$ is the outgoing unit vector. Note that this work treats $f_r(\mathbf{x}, L, O)$ and $f_r(L, O)$ 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 be reflected from the surface than was incident on it. 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 [56]. Finally, plausible BRDFs are positive — they take on

nonnegative values for all valid inputs [56]. A surface cannot reflect negative light, so this should feel natural. Explicitly, energy conservation is expressed with [56]:

$$\forall L \in \mathbb{S}^2 : \int_{O \in \mathbb{S}^2} f_r(L, O) d\mathbb{S}^2 \leq 1. \quad (2.17)$$

Eq 2.17 states that for all possible illumination directions L , integrating all possible outgoing observer directions O on the unit sphere cannot return greater than one from the energy conservation integral. Reciprocity can be formalized as:

$$\forall L, O \in \mathbb{S}^2 : f_r(L, O) = f_r(O \rightarrow L). \quad (2.18)$$

Now that the requirements for a plausible physical BRDF have been established, a collection of commonly-used BRDFs can be presented. The following BRDFs are all energy conserving, reciprocal, and nonnegative. Real-world materials are modeled with these BRDFs by fitting the free parameters of a given BRDF to measured reflectance data [28]. If a BRDF can be fit to the data with low error, that reflection function can be considered to be a faithful model for the true physics.

2.1.6.1 Lambertian

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

$$f_r(L, O) = \frac{C_d}{\pi} \quad (2.19)$$

In Eq 2.19, $0 \leq C_d \leq 1$ is the surface's coefficient of diffuse reflection. For example, $C_d = 0.4$ means that the surface reflects 40% of incident radiation and absorbs the other 60%.

2.1.6.2 Phong

While the diffuse BRDF reflects energy isotropically, many real-world reflections are highly biased. At the extreme end, a perfect mirror reflection is effectively a Dirac delta

function in the reflected illumination direction. Many real-world materials are well modeled as a linear combination of diffuse and specular effects. A simple specular BRDF model is that developed by Phong in 1975 [24]. 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 $n \geq 0$ [55].

$$f_r(L, O) = \frac{C_d}{\pi} + \frac{C_s \frac{n+2}{2\pi} (O \cdot R)^n}{N \cdot L} \quad (2.20)$$

In Eq 2.20, 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 perfect mirror reflection. Because of the introduction of a new coefficient of reflection, a new constraint is needed 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 explicit coefficient of absorption C_a which captures the fraction of incident radiation absorbed by the surface, yielding $C_d + C_s + C_a = 1$.

2.1.6.3 Blinn-Phong

The Blinn-Phong BRDF is similar to the Phong BRDF, but it parameterizes the specular lobe in terms of the halfway vector H [55]. 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.

$$f_r(L, O) = \frac{C_d}{\pi} + \frac{C_s \frac{n+2}{2\pi} (N \cdot H)^n}{4(N \cdot L)(N \cdot O)} \quad (2.21)$$

2.1.6.4 Glossy

The so-called glossy BRDF simulates reflections from plastic materials using an Gaussian distribution around the specular scattering lobe [55]:

$$f_r(L, O) = \frac{C_d}{\pi} + \frac{C_s}{2\pi\sigma^2(N \cdot L)} e^{\frac{-(R \cdot O)^2}{2\sigma^2}}, \quad (2.22)$$

where the parameter σ controls the width of the specular Gaussian.

2.1.6.5 Cook-Torrance

The Cook-Torrance explicitly accounts for the orientation of the microfacets making up the surface [26]. This model is built from a facet slope distribution term D_{ct} , a Fresnel term F , and a geometric attenuation factor G_{ct} . The slope distribution term describes the probability density of a given facet being oriented with a normal vector aligned with the halfway vector H [26]. A common formulation of this term is due to Beckmann:

$$D_{ct} = \frac{1}{\pi\alpha^2(N \cdot H)^4} \exp\left(\frac{1 - (N \cdot H)^2}{\alpha^2(N \cdot H)^2}\right). \quad (2.23)$$

In Eq 2.23, $\alpha \in [0, 1]$ is the roughness of the surface [26]. The Fresnel term accounts for the variation in specular reflection due to the angle of incidence. In reality, this term is a function of wavelength, but it is often approximated as [26]:

$$F = C_s + (1 - C_s)(1 - (H \cdot L))^5. \quad (2.24)$$

The geometric attenuation term expresses how microfacets shadow each other, and can be approximated with [26]:

$$G_{ct} = \min\left\{1, \frac{2(N \cdot H)(N \cdot O)}{O \cdot H}, \frac{2(N \cdot H)(N \cdot L)}{O \cdot H}\right\}. \quad (2.25)$$

The overall reflectance for the Cook-Torrance BRDF is given by:

$$f_r(L, O) = \frac{C_d}{\pi} + \frac{D_{ct} \cdot G_{ct} \cdot F}{4(N \cdot L)(N \cdot O)} \quad (2.26)$$

2.1.6.6 Oren-Nayar

The Oren-Nayar BRDF is an improved model of diffuse reflectance for many real-world materials like ceramics and the surface of the Moon. These materials diverge from the Lambertian model near the horizon, reflecting much more light than would be predicted by simple cosine loss [27]. This BRDF relies on the surface roughness α to compute a set of constants [27]:

$$\begin{aligned} A &= 1 - 0.5 \frac{\alpha^2}{\alpha^2 + 0.33} & (2.27) \\ B &= 0.45 \frac{\alpha^2}{\alpha^2 + 0.09} \\ C &= \frac{\text{proj}_N(L)}{\|\text{proj}_N(L)\|} \cdot \frac{\text{proj}_N(O)}{\|\text{proj}_N(O)\|} \\ \beta &= \max \{\arccos(L \cdot N), \arccos(O \cdot N)\} \\ \gamma &= \min \{\arccos(L \cdot N), \arccos(O \cdot N)\}. \end{aligned}$$

With these values computed, the reflectance of the Oren-Nayar BRDF is given by [27]:

$$f_r(L, O) = \frac{C_d}{\pi} (A + (B \max \{C, 0\} \sin \beta \tan \gamma)). \quad (2.28)$$

2.1.6.7 Ashikhmin-Shirley

The Ashikhmin-Shirley BRDF is unique among those presented in this section as it allows for anisotropic reflection which can be non-negligible for metals [25]. This model is parameterized by the diffuse reflection coefficient, as well as two specular exponents n_u and n_v . When $n_u = n_v$, the model behaves much like the Phong BRDF. The full BRDF is expressed [25]:

$$\rho_d = \frac{28C_d}{23\pi} (1 - C_s) \left(1 - \left(1 - \frac{N \cdot L}{2}\right)^5\right) \left(1 - \left(1 - \frac{N \cdot O}{2}\right)^5\right) \quad (2.29)$$

$$\rho_s = \frac{\sqrt{(n_u + 1)(n_v + 1)}}{8\pi} \frac{(N \cdot H)^{\frac{n_u(H \cdot U)^2 + n_v(H \cdot V)^2}{1 - (H \cdot N)^2}}}{(H \cdot L) \max\{N \cdot O, N \cdot L\}} F$$

$$f_r(L, O) = \rho_d + \rho_s.$$

In Eq 2.29, F is the same Fresnel factor in Eq 2.24 while U and V are predefined surface basis vectors perpendicular to the surface normal.

2.1.7 BRDF Summary

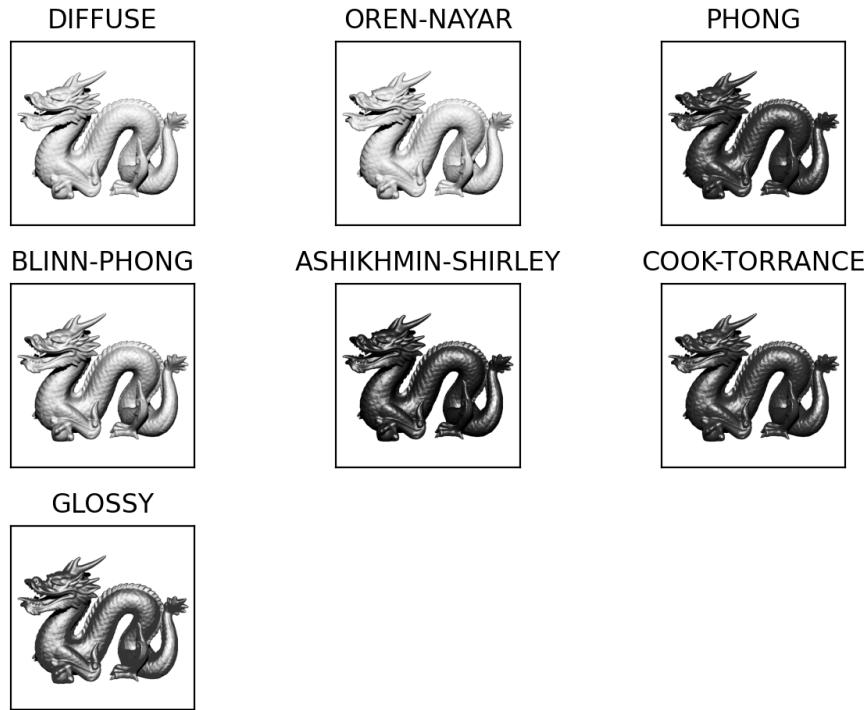


Figure 2.2. Implemented BRDFs rendered with arbitrary parameters, demonstrating the qualitative differences between lighting models

Material	C_d	C_s	n
Solar panel (Phong) [57]	0.15	0.25	0.26
Bus structure (Phong) [57]	0.34	0.40	8.9
Multilayer insulation (MLI) (Phong)	0.1	0.9	20
White paint (Phong)	0.9	0.1	1

Table 2.1. Reflection properties of real materials. MLI and white paint parameters are estimated qualitatively by the author. Significant variance may exist between objects depending on surface finishes [28].

2.1.8 Real Material Properties

Producing accurate light curves requires accurate optical properties for the simulated materials. Matusik et al. compiled measured bidirectional reflectance data from over 100 real materials including various metals and paints that might be found on human-made space objects [28]. Optimally, such a dataset would be used to model the structure, solar panels, and antenna surfaces of the simulated objects. However, it is far more computationally efficient to fit one of the presented three parameter BRDFs to observed data. The Phong model parameters used for simulations in this work are listed in Table 2.1.

2.1.9 Brightness Units

2.1.9.1 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 $10 [m^2]$ area would experience $10 [W]$ of incident power. The mean irradiance of the Sun — termed the solar constant — is derived from the mean luminosity of the Sun $L_s = 3.828 \cdot 10^{26} [W]$ and a distance of 1 Astronomical Unit $1 [AU] = 149597870.7 [km]$ [4]. The mean solar constant is then computed via

$$I_s = \frac{L_s}{4\pi(AU)^2} \approx 1361.0 \left[\frac{W}{m^2} \right]. \quad (2.30)$$

In reality, the exoatmospheric irradiance at Earth varies [4]. Figure 2.3 displays the small variations in the solar irradiance due to the solar cycle and the much larger variations due to Earth’s eccentric orbit.

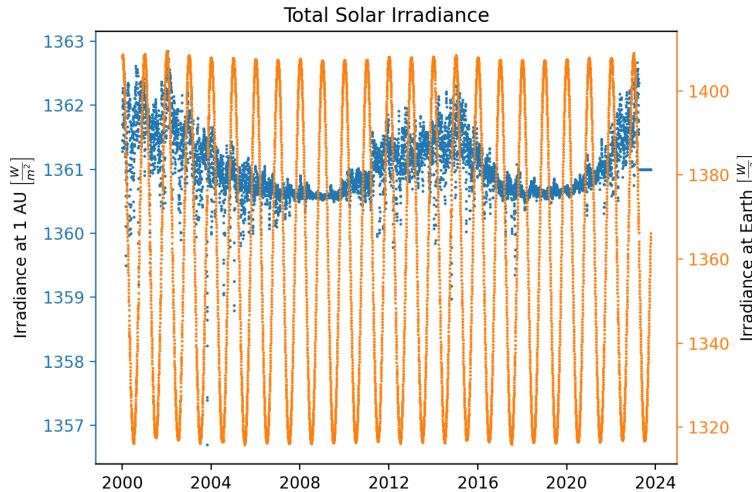


Figure 2.3. Total solar irradiance variations at Earth and 1 AU. The solar cycle is apparent when distance is held constant at 1 AU, yielding small variations in the peak intensity at Earth.

2.1.9.2 Apparent Magnitude

Apparent magnitude — also known as visual or relative magnitude — is a reverse logarithmic scale that originates in astronomy [4]. 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 [4]. Apparent magnitude m is computed from irradiance via [4]:

$$m = -2.5 \log_{10} \left(\frac{I}{I_0} \right). \quad (2.31)$$

In Eq 2.31, I is the irradiance of the source of interest and I_0 is irradiance of the zero-point source. This makes sense; substituting $I = I_0$ returns $m = 0$. The star Vega is usually taken to be the zero-point with irradiance $I_0 = 2.518021002 \cdot 10^{-8} \left[\frac{W}{m^2} \right]$ [4].

Eq 2.31 is rearranged to compute irradiance from a given apparent magnitude, yielding:

$$I = I_0 \cdot 10^{-\frac{m}{2.5}}. \quad (2.32)$$

2.1.9.3 Normalized Irradiance

The light curve simulation methods presented in this work use normalized irradiance, the received irradiance of a object at a distance of 1 meter per $[W/m^2]$ of the distant source illuminating it. This is a non-standard quantity in the literature, but proves useful for the same reasons absolute magnitude is used by astronomers. Adjusting the signal to a standard distance and illumination strength enables the simulation and inversion of light curves in a non-dimensional environment. This simplifies simulation and makes the shape 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 due to the Sun \hat{I} via:

$$\hat{I} = \frac{r^2}{I_s} I. \quad (2.33)$$

To be precise, the normalized irradiance has units m^2 and can be interpreted as the area of a flat plate illuminated by a 1 $[W/m^2]$ light source required to reflect the same irradiance in the direction of the observer. Throughout this work, normalized irradiance is treated as being nondimensional for light curve simulation and inversion purposes as its physical interpretation is largely irrelevant.

2.1.10 Simulating Normalized Light Curves of Convex Objects

Light curve simulation for convex geometry can be solved semi-analytically as each face's contribution to the measured irradiance can be computed individually [10]. Determining whether a face is illuminated requires two horizon checks to determine visibility from the

Sun and to the observer. For a face i at timestep j these horizon checks are expressed by the shadowing condition μ_{ij} .

$$\mu_{ij} = \begin{cases} 1 & \text{if } (O_j \cdot \hat{n}_i) > 0 \text{ and } (L_j \cdot \hat{n}_i) > 0 \text{ and } \delta_{ij,ss} = 0 \text{ and } \delta_{ij,os} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.34)$$

The unit vectors O and L point from the center of mass of the object to the observer and Sun, respectively. The outward-pointing face normal unit vector \hat{n} is chosen by convention for all mesh operations. The self-shadowing and observer-shadowing conditions, $\delta_{ij,ss}$ and $\delta_{ij,os}$, are always zero for convex polyhedra but are crucial for accurately simulating nonconvex 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 faces, where the BRDF is evaluated at each face. Each contribution is expressed as the product of the normalized irradiance \hat{I} . This can be scaled to adjust for the distance from the observer to the object and the solar constant to yield the noiseless received irradiance. For any physically plausible BRDF, the normalized light curve for a convex object is expressed as a sum over all faces i at time step j [5]:

$$\hat{I}_j = \sum_{i=1}^m \mu_{ij} a_i f_r(L_j \rightarrow O_j) (\hat{O}_j \cdot \hat{n}_i) (\hat{S}_j \cdot \hat{n}_i). \quad (2.35)$$

In Eq 2.35, a_i is the area of the i th face. Note that $f_r(L_j \rightarrow O_j)$ must be computed separately for each face to account for normal vector and coefficient of reflection differences. This normalized light curve can be dimensionalized by the total solar irradiance and the distance r_j between the observer \mathbf{r}_{obs} to the object center of mass \mathbf{r}_{obj} :

$$I_j = \frac{I_s \hat{I}_j}{r_j^2}. \quad (2.36)$$

Note that using $r_j = \|\mathbf{r}_{obj} - \mathbf{r}_{obs}\|$ is a simplifying assumption as some faces of the object are further or closer to the Sun than the center of mass. As the scale of human-made space objects are a factor of 10^{10} to 10^{12} smaller than the distance to the Sun, the difference across the surface of the object is negligible.

2.1.11 Simulating Normalized Light Curves of Nonconvex Objects

The light curve simulation framework developed in this work relies on a pixel-by-pixel image rendering on the GPU to compute the total reflected irradiance at each timestep of the simulation. As a result, a background in common computer graphics terminology and reference frames is necessary to understand the transformation from coordinates in the object body frame to a pixel in the final image.

2.1.11.1 Camera Projections and Terminology

In computer graphics, a camera is defined by its position R_{cam} , target T_{cam} , reference up direction U_{cam} , and a field of view FOV . In order to render a 3D scene to a 2D image using this camera, a transformation is needed. An orthographic camera accomplishes this transformation by orthogonally projecting points onto a plane perpendicular to camera look direction $T_{cam} - R_{cam}$. The volume of space that falls into view of the camera is known as the frustum [58]. An important consequence of orthogonal projection is that objects further away from the camera do not shrink in the image, which is sometimes advantageous. A perspective projection forms a more complex frustum that expands outwards from a near plane to a proportionally larger far plane. Because of the expansion of the frustum, geometry processed by the perspective transformation shrinks as it moves further from the near plane. Figure 2.4 displays the geometry of these camera frustums, with additional reference frames detailed in Section 2.1.11.2.

While the perspective projection is shown in Figure 2.4, only orthographic projection is relevant for future simulation tasks as both the observer and sun are assumed to be very far from the object. In the limit, the true perspective nature of the camera behaves as an

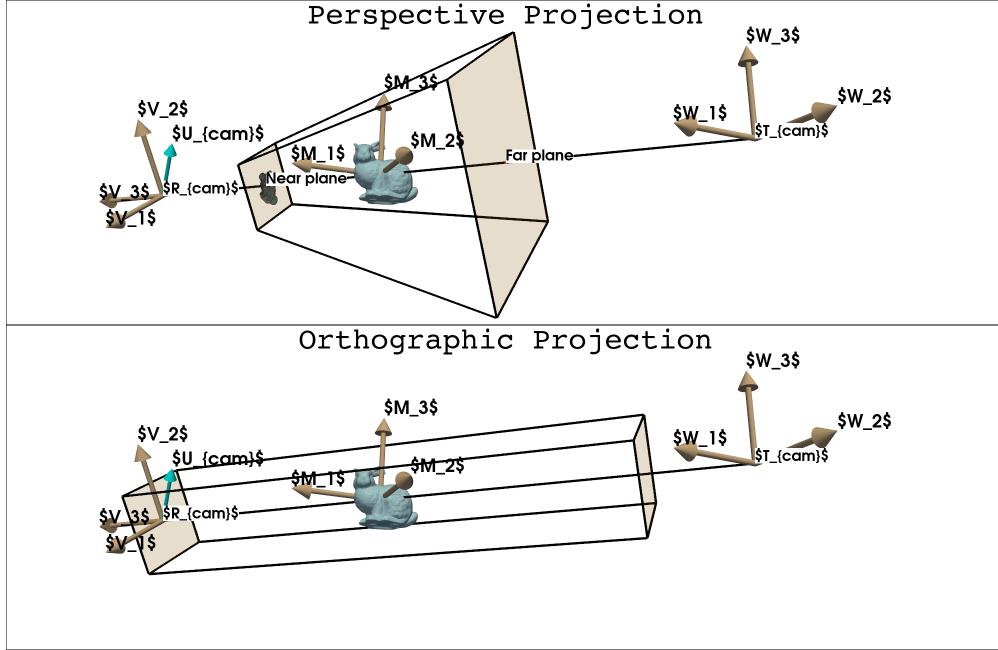


Figure 2.4. Perspective and orthographic projections with relevant camera frustum attributes labeled

orthographic projection as negligible foreshortening occurs for objects as they retreat from the camera.

2.1.11.2 Transformations

Transformations in computer graphics are often represented by 4×4 matrices. These matrices operate on so-called homogeneous coordinates, operating on vectors in \mathbb{R}^4 of the form $[x, y, z, w]^T$ [58]. The inclusion of the fourth component enables simultaneous rotation, translation, and scaling of the input points by the matrix, provided that the outputs are normalized to set $w = 1$ [58].

There are a few fundamental matrices used in computer graphics that are necessary for later hardware-accelerated light curve simulation algorithms. The first is the model matrix $M \in \mathbb{R}^{4 \times 4}$ which transforms from the world frame to the model frame given the origin of the model body frame $R_m \in \mathbb{R}^{3 \times 3}$ and the orientation of the model body frame relative to the world frame $\mathbf{q}_m \in \mathbb{R}^4$ as a quaternion [58]:

$$M = \begin{bmatrix} -q_2^2 - q_3^2 + q_1^2 + q_4^2 & 2q_1q_2 + 2q_3q_4 & 2q_1q_3 - 2q_2q_4 & -R_{m,x} \\ 2q_1q_2 - 2q_3q_4 & -q_1^2 - q_3^2 + q_2^2 + q_4^2 & 2q_1q_4 + 2q_2q_3 & -R_{m,y} \\ 2q_1q_3 + 2q_2q_4 & 2q_2q_3 - 2q_1q_4 & -q_1^2 - q_2^2 + q_3^2 + q_4^2 & -R_{m,z} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.37)$$

Eq 2.37 uses Eq 2.7 to transform the quaternion into a DCM. The model matrix basis vectors are illustrated in Figure 2.4 as M_i , with the world frame basis vectors notated with W_i . Given the location of the camera origin $R_{cam} \in \mathbb{R}^3$, its target position $T_{cam} \in \mathbb{R}^3$, and the camera up direction $U_{cam} \in \mathbb{R}^3$, the view matrix $V \in \mathbb{R}^{4 \times 4}$ that transforms from the world frame to the camera frame is given by [58]:

$$v_3 = \frac{R_{cam} - T_{cam}}{\|R_{cam} - T_{cam}\|} \quad (2.38)$$

$$v_1 = \frac{U_{cam} \times v_3}{\|U_{cam} \times v_3\|}$$

$$v_2 = v_1 \times v_3$$

$$V = \begin{bmatrix} v_{1,x} & v_{1,y} & v_{1,z} & -v_1 \cdot R_{cam} \\ v_{2,x} & v_{2,y} & v_{2,z} & -v_2 \cdot R_{cam} \\ v_{3,x} & v_{3,y} & v_{3,z} & -v_3 \cdot R_{cam} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The view matrix basis vectors are illustrated in Figure 2.4 as V_i . Given the field of view of the camera FOV in radians, the distance from the camera origin to the near n and far f clipping planes, and the camera aspect ratio a , the orthographic projection matrix $P \in \mathbb{R}^{4 \times 4}$ that transforms from the camera frame to the image plane is given by [58]:

$$t = n \cdot \tan\left(\frac{FOV}{2}\right) \quad (2.39)$$

$$r = t \cdot a$$

$$P = \begin{bmatrix} \frac{2n}{2r} & 0 & 0 & 0 \\ 0 & \frac{2n}{2t} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & \frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Together, these matrices form the so-called Model-View-Projection matrix which transforms directly from coordinates in the object body frame r_{obj} to the image plane:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ w_h \end{bmatrix} = MVP \begin{bmatrix} r_{obj,x} \\ r_{obj,y} \\ r_{obj,z} \\ 1 \end{bmatrix}. \quad (2.40)$$

In Eq 2.40, x_h/w_h and y_h/w_h are homogeneous coordinates in the image plane, running linearly from $[-1, -1]$ at the top left corner of the image to $[1, 1]$ at the bottom right. Given the width of the image in w_{pix} pixels, the image coordinates (x, y) of r_{obj} in pixels are:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} w_{pix} \left(\frac{1}{2} + \frac{x_h}{2w_h} \right) \\ a \cdot w_{pix} \left(\frac{1}{2} + \frac{y_h}{2w_h} \right) \end{bmatrix}. \quad (2.41)$$

The transformation summarized in Eqs 2.40 and 2.41 are crucial in Section 2.1.11.4 for the shadow mapping algorithm used to compute pixel-wise self-shadowing effects.

Many existing light curve simulation methods for nonconvex objects rely on ray tracing schemes like Möller and Trumbore's ray-triangle intersection algorithm [5], [23]. This computation has complexity $\mathcal{O}(n^2)$ if implemented naïvely, but can be improved to $\mathcal{O}(n \ln n)$ with better spatial data structures. For human-made space objects, there may be significant self-shadowing at large phase angles. As a result, it cannot be assumed that the self-shadowing conditions $\delta_{ij,ss}$ and $\delta_{ij,os}$ are zero [5], [14]. Naïve ray traced shadows generally

require $\mathcal{O}(n^2)$ ray-triangle intersections per timestep for n faces. For this reason, ray traced shadows quickly become infeasible for complex objects without GPU parallelization. The limitations of ray-triangle intersections for light curve simulation is discussed at length by Frueh et al. [14].

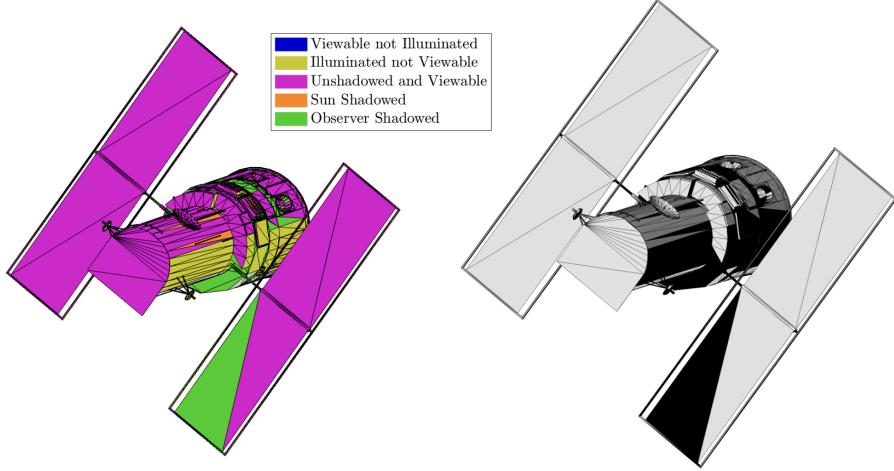


Figure 2.5. Hubble Space Telescope ray traced shadow categorization and shading. Models from [59]

The facet-wise shadow categorization shown in Figure 2.5 informs the μ_{ij} self-shadowing term in Eq 2.35. The major downside of facet-wise shadow categorization is that it is difficult to determine the fraction of each face is illuminated – it categorizes the entire face together, leading to unphysical discontinuities in simulated light curves.

2.1.11.3 The Importance of Self-Shadowing

To motivate the need for accurate shadows when dealing with 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 2.6 displays light curves for the asteroid Bennu and the Hubble Space Telescope with and without accurate shadows under a single-axis spin profile with inertially

fixed Sun and observer vectors. Without accurate shadowing, the light curve's intensity and its time derivative can be significantly error-prone.

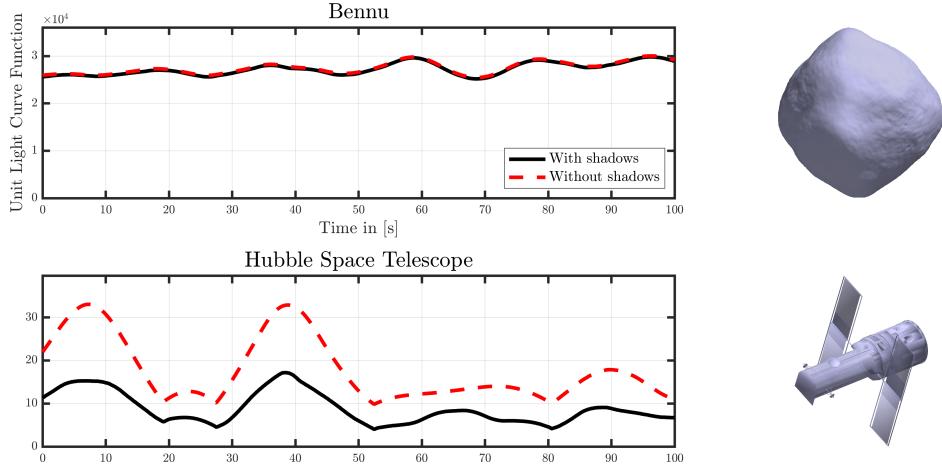


Figure 2.6. Normalized irradiance errors introduced by neglecting shadows for Bennu and HST. Models from [59]

2.1.11.4 Shadow Mapping

Shadow mapping is used in the simulations presented in this work for faster and more accurate self-shadowing. Shadow mapping is a well understood technique in computer graphics [60]. Although modern ray traced shadowing may be more computationally efficient, shadow mapping was selected for its ease of implementation [60]. Shadow mapping shades individual pixel fragments instead of entire faces, offering increasing shadow quality over facewise ray tracing as the number of mesh faces falls.

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, where a second depth texture is computed. This second texture is used to find the closest fragment along each ray to the Sun [61]. Self-shadowed fragments are classified as those further from the Sun than the closest fragment along the same ray, indicated in red in Figure 2.7. Fragments that do not pass the convex shadowing condition are horizon shadowed, indicated in blue in Figure 2.7, determining the

Sun and observer shadowing conditions at once. All remaining fragments are shaded with using the same Lambertian reflection model in 2.35. 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 [62].

In order to compute the final shaded and shadowed image, a depth map must be computed from the perspective of two orthographic cameras in the Sun and observer directions. These depth masks require a set of transformations from the model body frame to screen space. With this background laid out, the process for computing a perpendicular depth map $d(x, y)$ from the perspective of an arbitrary orthographic camera is detailed in Algorithm 1.

Algorithm 1 Pixel-wise depth map computation for shadow mapping

$(x, y) \in \mathbb{Z}$	▷ Pixel coordinates on the image plane
$R_{pix} \in \mathbb{R}^3$	▷ Pixel world coordinates; provided by OpenGL
$d(x, y) \leftarrow (R_{cam} - R_{pix}) \cdot R_{cam}$	▷ Pixel depth in the camera view direction

The pixel-wise shading process is summarized in Algorithm 4 in Appendix 7.1.1. This shading and shadow mapping procedure is detailed graphically in Figure 2.7.

An image shaded by Algorithm 4 is dimensionalized to a normalized irradiance value by computing the area of the square orthographic field of view given the image pixel count n_{pix} :

$$A_{image} = 2n_{pix} (FOV)^2. \quad (2.42)$$

This field of view must be set dynamically at runtime to account for the size of the 3D model. The pixel values v in the image are stored as unsigned 8 bit integers which take on values from 0 to 255, corresponding to the power range 0 to 1 within the pixel shading procedure described by Algorithm 4. The total normalized irradiance in image j is computed by summing over all n_{pix} pixels and dimensionalizing by the area of the image plane:

$$\hat{I}_j = \frac{A_{image}}{n_{pix}} \sum_{i=1}^{n_{pix}} \frac{v_i}{255} \quad (2.43)$$

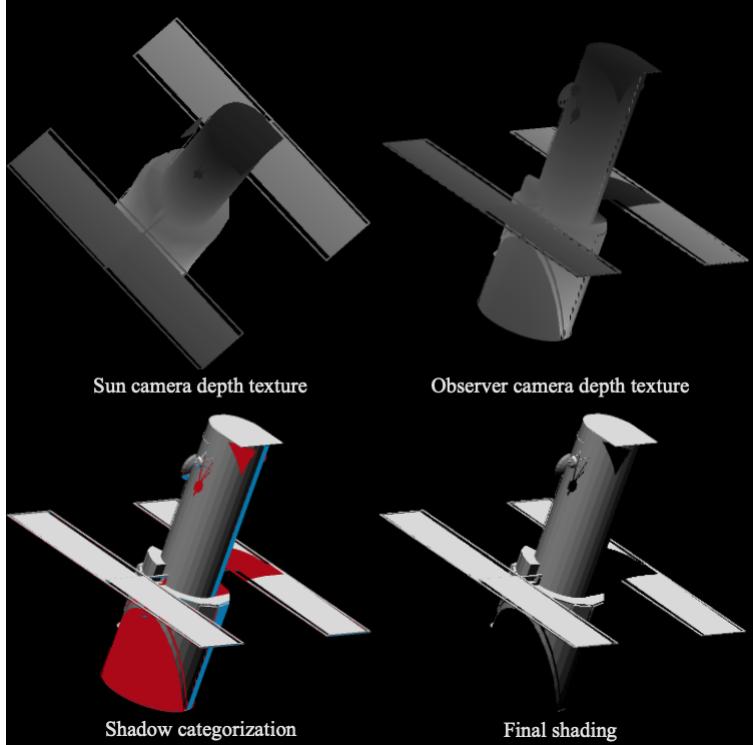


Figure 2.7. Hubble Space Telescope shadow mapping with self (red) and horizon (blue) shadows rendered. Models from [59]

2.2 Simulating The Observations

2.3 Time Systems

In order to accurately predict the position of a space object and an observer, it is necessary to understand how a given time translates to the orientation of the relevant reference frames. These calculations require conversions between various time scales, the Julian date, and sidereal time.

2.3.1 Time Scales

There are a variety of scales used to measure time. What follows is a minimal treatment of each. For a more comprehensive overview, see Section 3.5 of [3]. International Atomic Time (TAI) is based on measurements from atomic clocks and is independent of astronomical

effects or observations. By definition, TAI proceeds at the rate of 1 SI second per second. 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 $TT - TAI = 32.184$ seconds from TAI and proceeding at the same rate as TAI. These time scale relations are summarized in Eq 2.44.

$$UTC = UT1 - \Delta UT1 \quad (2.44)$$

$$TAI = UTC + \Delta AT$$

$$TT = TAI + 32.184^s$$

These time scales are relevant for this research as the precise coordinate frame transformation from ITRF to the J2000.0 realization of ICRF relies on quantities expressed in UT1. Date timestamps are usually standardized to UTC, requiring the transformations in Eq 2.44 for full accuracy. Figure 2.8 shows the evolution of UTC, UT1, and TT with respect to TAI. Notice that $\Delta UT1$ continually changes while ΔAT is always truncated to a nearby integer.

2.3.2 Julian Date

Most tasks in astrodynamics are easier when using a continuous time system. For this reason, the Julian date is adopted. This quantity is defined as the number of days elapsed since January 1, 4713 B.C., at 12:00 [3]. Given a date timestamp of the form D/M/Y h:m:s between the years of 1900 and 2100, the Julian date is computed via:

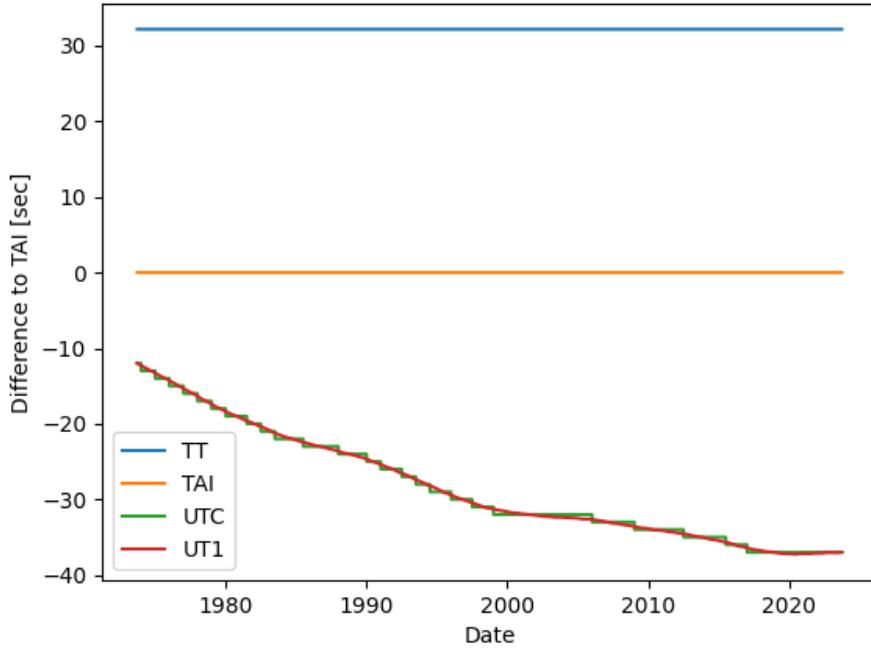


Figure 2.8. Time scales relative to TAI

$$JD = 376Y - \text{floor} \left[\frac{7Y + 7 \cdot \text{floor} \left(\frac{M+9}{12} \right)}{4} \right] + \text{floor} \left(\frac{275M}{9} \right) + d + 1721013.5 + \frac{\frac{(s+60)}{60} + h}{24}. \quad (2.45)$$

Note that Eq 2.45 is always a function of the time scale used in the input, i.e. a UTC timestamp yields JD_{UTC} whereas a UT1 timestamp yields JD_{UT1} . Another useful quantity for later time and coordinate system calculations is the number of Julian centuries since a particular epoch. The J2000.0 epoch is used unless otherwise stated, resulting in [3]:

$$T = \frac{JD - 2451545.0}{36535}. \quad (2.46)$$

Often, more specificity is needed with respect to the time scale used in Eq 2.46. For example, computing T with an input date in UT1 yields T_{UT1} using JD_{UT1} , which is in turn a function a date timestamp expressed in UT1.

2.3.3 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 [3]. By contrast, a sidereal day is the time required for the stars to complete a revolution around an observer's meridian. Due to the Earth's orbit around the Sun, the sidereal day is about 4 minutes shorter than the solar day [3]. The Greenwich mean sidereal time (GMST) is computed in seconds via [4]:

$$\theta_{GMST} = 67310.54841 + \left(3.15576 \cdot 10^9 + 8640184.812866 \right) T_{UT1} + 0.093104 T_{UT1}^2 - 6.2 \cdot 10^{-6} T_{UT1}^3. \quad (2.47)$$

Accounting for the variations in the inclination of the ecliptic ϵ and the the change in the equinox compared to the reference epoch $\Delta\Psi$ produces Greenwich apparent sidereal time (GAST) via [4]:

$$\theta_{GAST} = \theta_{GMST} + \Delta\Psi \cos \epsilon. \quad (2.48)$$

Both the inclination of the ecliptic and the difference in the equinox are computed with series expansions following the IAU 1980 theory of nutation [3].

2.4 Coordinate Systems

A precise definition of coordinate systems is necessary for determining the position of the observer and the observed space object at a given time. The relevant conversions are summarized by a single question: how is a fixed position on the surface of the Earth transformed into a standardized inertial reference frame?

2.4.1 Latitude, Longitude and Altitude

Latitude, longitude, and altitude (LLA) is a spherical representation of position on or above the surface of the Earth. For the purposes of precise station positioning, the difference between the two types of longitude — geocentric and geodetic — is important. Geocentric latitude is the angle between the line from the center of mass of the Earth to the position of interest and the equatorial plane. Geodetic latitude instead measures the angle between the local ellipsoid surface normal and the equatorial plane. Geodetic latitude ϕ_{geod} is converted to geocentric ϕ_{geoc} latitude with [4]:

$$\phi_{\text{geoc}} = \tan^{-1} \left((1 - f)^2 \tan \phi_{\text{geod}} \right). \quad (2.49)$$

Additionally, the radius of the ellipsoid r_E at a given geocentric latitude is necessary for later conversion, expressed by [4]:

$$r_E = R_E - f \sin^2(\phi_{\text{geoc}}). \quad (2.50)$$

The altitude in an set of LLA coordinates needs a reference point. Different observers may be defined differently — either relative to the approximate ellipsoidal shape of the Earth, the hypothetical mean sea level, or the surrounding terrain.

2.4.1.1 Ellipsoid

Due to Earth's equatorial bulge, it is common to model the rough shape of the Earth as an ellipsoid. In particular, the 1984 World Geodetic Survey (WGS-84) model is used throughout this work to define the shape of the Earth ellipsoid, with parameters listed in Table 2.2 for use in Eqs 2.49 and 2.50.

Parameter	Value
Equatorial radius R_E	6378.137 [km]
Flattening ratio f	1/298.257

Table 2.2. WGS-84 ellipsoid model of the Earth [3].

These parameters are needed for the conversion from LLA to the International Terrestrial Reference Frame.

2.4.1.2 Geoid

The geoid accounts for the gravitational potential differences across the Earth's surface [3]. It is a surface of equal gravitational potential; the surface the ocean relaxes to without the influence of the wind and tides [3]. For this reason, the geoid is alternatively known as the mean sea level (MSL). The ellipsoid is a good approximation of the geoid, which deviates from the ellipsoid by less than ≈ 100 meters at all latitudes and longitudes. The height of the geoid above the ellipsoid can be computed from a high-fidelity gravity model, but it is often more convenient to interpolate a pre-computed grid of geoid heights. Figure 2.9 displays global geoid heights derived from the 1996 Earth Gravitational Model (EGM-96) relative to the ellipsoid.

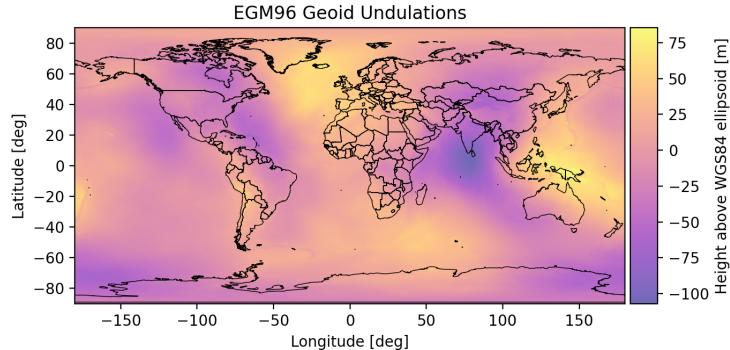


Figure 2.9. EGM-96 geoid heights above the WGS-84 ellipsoid

2.4.1.3 Terrain

Terrain elevation is usually the final component needed to fully define the altitude of a ground station, which is often defined relative to MSL. This work uses 30-meter terrain tiles from the Shuttle Radar Topography Mission (SRTM). Figure 2.10 shows the local elevation around the Purdue Optical Ground Station using SRTM data.

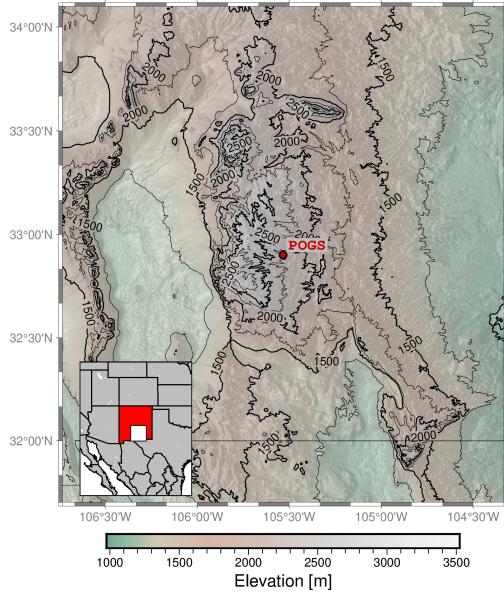


Figure 2.10. MSL elevations surrounding the Purdue Optical Ground Station

2.4.1.4 Altitude Conversions

Given an altitude relative to the terrain a_{terrain} , the elevation above the ellipsoid a_{ellip} is given as a function of the terrain elevation above the geoid $h_{\text{terrain}}(\lambda, \phi)$ and the geoid elevation above the ellipsoid $h_{\text{geoid}}(\lambda, \phi)$ by [3]:

$$a_{\text{ellip}} = a_{\text{terrain}} + h_{\text{terrain}}(\lambda, \phi_{\text{geod}}) + h_{\text{geoid}}(\lambda, \phi_{\text{geod}}). \quad (2.51)$$

2.4.2 International Terrestrial Reference Frame (ITRF)

The Cartesian form of LLA is known as the Earth-centered Earth-fixed (ECEF) reference frame. Throughout this work, ECEF and ITRF will be used interchangeably. This 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 — orienting 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 with

$\hat{y} = \hat{z} \times \hat{x}$ yields a reference frame that remains fixed, neglecting effects like continental drift.

The transformation from LLA ($\lambda, \phi_{\text{geod}}, a_{\text{ellip}}$) to ITRF is given by [3]:

$$\begin{aligned} e^2 &= 2f - f^2 & (2.52) \\ N &= \frac{R_E}{\sqrt{(1 - e^2 \sin(\phi_{\text{geod}})^2)}} \\ \rho &= (N + a_{\text{ellip}}) \cos(\phi_{\text{geod}}) \\ x_{\text{itrif}} &= \rho \cos(\lambda) \\ y_{\text{itrif}} &= \rho \sin(\lambda) \\ z_{\text{itrif}} &= (N(1 - e^2) + a_{\text{ellip}}) \sin(\phi_{\text{geod}}). \end{aligned}$$

In Eq 2.52, e^2 is the squared eccentricity of the ellipsoid, N is the radius of curvature in the meridian, and ρ is the $x - y$ plane magnitude of the station's position [3].

Many later transformations require the body axis rotation matrices R_1 , R_2 , and R_3 which are expressed:

$$\begin{aligned} R_1(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} & (2.53) \\ R_2(\theta) &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \\ R_3(\theta) &= \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

2.4.3 Topocentric East North Up (ENU) Reference Frame

The remaining transformations in this chapter will only be defined in terms of their rotation matrices. It is often useful to express observations in a local reference frame. The East North Up (ENU) coordinate system is used throughout this work. This system has an origin at the observing station, with the first two basis vectors pointing towards the local East and North and the third pointing towards zenith. The transformation from a vector \vec{r}_{itr} in ITRF to \vec{r}_{enu} in ENU is given by [4]:

$$\vec{r}_{\text{enu}} = F_2 F_1 R_2(\phi_{\text{geoc}}) R_3(\lambda) \vec{r}_{\text{itr}}. \quad (2.54)$$

In Eq 2.54, R_3 is a rotation about the third body axis, F_1 swaps the second and third unit vectors, and F_2 swaps the first and third unit vectors. The orientation of the ENU reference frame at the Purdue Optical Ground Station is depicted in Figure 2.11.



Figure 2.11. ENU reference frame orientation at Purdue Optical Ground Station

2.4.4 International Celestial Reference Frame (ICRF/J2000)

Transforming from ITRF to the a standardized inertial reference frame is an involved process due to the variety of nonlinear effects impacting the Earth's rotational motion. In

total, this transformation must account for polar motion, the nutation and precession of the Earth's pole, and the mean sidereal time. These transformations are treated much more thoroughly in Montenbruck and Gill [63].

Accounting for polar motion — the motion of the Earth's pole that cannot be explained through nutation theory — transforms from \vec{r}_{itr} in ITRF to \vec{r}_{gtod} in Greenwich True of Date (GTOD) via [3]:

$$\vec{r}_{\text{gtod}} = R_1(y_p)R_2(x_p)\vec{r}_{\text{itr}}, \quad (2.55)$$

where x_p and y_p are the angular components of the polar motion at the time of interest [4]. Accounting for the sidereal rotation of the Earth about its pole transforms from \vec{r}_{gtod} in GTOD to \vec{r}_{teme} in the True Equator, Mean Equinox (TEME) reference frame via [4]:

$$\vec{r}_{\text{teme}} = R_3(-\theta_{\text{GMST}})\vec{r}_{\text{gtod}}, \quad (2.56)$$

where θ_{GMST} is the Greenwich mean sidereal time. Accounting for the difference between GMST and GAST at the date of interest transforms \vec{r}_{teme} TEME to \vec{r}_{tod} in the True of Date (TOD) reference frame via [3]:

$$\vec{r}_{\text{tod}} = R_3(-\Delta\Psi \cos \epsilon)\vec{r}_{\text{teme}}, \quad (2.57)$$

where $\Delta\Psi$ is the angular difference in the longitude of the vernal equinox between GMST and GAST, and ϵ is the true inclination of the ecliptic [3]. Accounting for the nutation of Earth's pole transforms \vec{r}_{tod} from TOD to \vec{r}_{mod} in the Mean of Date (MOD) reference frame via [3]:

$$\vec{r}_{\text{mod}} = R_1(-\bar{\epsilon})R_3(\Delta\Psi)R_1(\bar{\epsilon} + \Delta\epsilon)\vec{r}_{\text{tod}}, \quad (2.58)$$

where $\bar{\epsilon}$ is the mean inclination of the ecliptic at the time of interest, and ϵ is the true inclination of the ecliptic [3].

Accounting for the secular precession of Earth's pole transforms \vec{r}_{mod} from MOD to \vec{r}_{icrf} in ICRF via [3]:

$$\vec{r}_{\text{icrf}} = R_3(\zeta)R_2(\theta)R_3(z)\vec{r}_{\text{mod}}, \quad (2.59)$$

through the 3-2-3 Euler angle sequence (z, θ, ζ) . Each of these angles modeling the precession are a function of the date of the transformation, and are approximated by polynomial expansions in the Julian century T [4].

The specific realization of ICRF used in this work is referenced to the position of the equator and equinox at the J2000.0 epoch (January 1, 2000 12:00:00.000 TT), leading to the common name for this reference frame, "J2000" [3].

2.4.5 Right Ascension and Declination

Right ascension and declination, often shortened to RA/Dec, are useful angles for describing the angular position of an object on the celestial sphere from the perspective of an observer. As such, RA/Dec values are inertial, but can be defined relative to a topocentric observer or the center of the Earth [4]. The topocentric RA/Dec is exclusively used in this work for later simulations. 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 $\hat{v} = [x_{\text{itrif}}, y_{\text{itrif}}, z_{\text{itrif}}]^T$ in J2000, RA/Dec is computed via [4]:

$$\begin{bmatrix} \alpha \\ \delta \end{bmatrix} = \begin{bmatrix} \text{atan2}(y_{\text{itrif}}, x_{\text{itrif}}) \\ \text{atan2}(z_{\text{itrif}}, \sqrt{x_{\text{itrif}}^2 + y_{\text{itrif}}^2}) \end{bmatrix}. \quad (2.60)$$

2.4.6 Azimuth and Elevation

Azimuth and elevation, often shortened to Az/El, are similar angular quantities to right ascension and declination [4]. 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 topocentric 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 2.60 can be repurposed in terms of Az/El, where $\hat{v} = [x_{ENU}, y_{ENU}, z_{ENU}]^T$ is expressed in the frame of interest [4].

$$\begin{bmatrix} Az \\ El \end{bmatrix} = \begin{bmatrix} \text{atan2}(y_{ENU}, x_{ENU}) \\ \text{atan2}(z_{ENU}, \sqrt{x_{ENU}^2 + y_{ENU}^2}) \end{bmatrix} \quad (2.61)$$

Note that Eq 2.61 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.

2.4.7 Coordinate Transformations Summary

With these transformations, any topocentric observation directions in RA/Dec or Az/El can be converted into a Cartesian representation and transformed into J2000. Simultaneously, any propagated space object orbits can be transformed similarly from an arbitrary propagation frame such that all future computations requiring the observation geometry take place in the same reference frame.

2.4.8 Planetary Ephemerides

The Spacecraft, Planet, Instrument, C-matrix, and Events (SPICE) toolkit is used for the propagation of the Sun and moon throughout the results presented in this work [64]. SPICE is maintained by the NASA Jet Propulsion Laboratory's Navigation and Ancillary Information Facility (NAIF) and provides the state of the art solutions for the positions and orientations of solar system bodies [64]. All calculations of \mathbf{r}_{Sun} and \mathbf{r}_{moon} rely on queries to SPICE using the DE440s planetary ephemeris files.

2.5 The Charge Coupled Device (CCD)

Many optical telescopes, including the Purdue Optical Ground Station, use a CCD to convert incident photons into a digital signal on a pixel grid [9]. CCDs accomplish this with a matrix of semiconductor pixel wells which collect electrons released by the photoelectric effect when photons are incident on that pixel [9]. The release of these photoelectrons is wavelength dependent and is captured by the quantum efficiency spectrum of a given CCD. Developing this background in photometry is useful for both light curve simulation as well as simulating the CCD camera measuring that light curve.

Due to the sheer distance from the observer to any space object, taking an image that resolves the object's geometry is generally impossible. For an observer with the pixel scale of the Purdue Optical Ground Station ($s_{pix} = 0.72$ [arcsec/pixel]), a 6U CubeSat in LEO occupies about 1/44 of a pixel area, while a large communications satellite in GEO occupies approximately 1/13 of a pixel area. In either case, no geometric information is left on the image plane even before diffraction is taken into account.

Whenever an optical telescope is observing an unresolved space object, the object's signal is necessarily superimposed on whatever signals exist in the background as the unresolved signal spreads much farther than the object's actual geometric bounds. In this context, background does not only refer to sources physically further than the object — as light can easily enter optical path through atmospheric scattering — but all sources that impact the image apart from the object signal. Some of these sources even originate within the telescope optics and its sensor. 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. The overall noise floor can be broken up into background signal sources and sensor effects.

2.5.1 Diffraction

Many objects of interest are far past LEO, making optical observations diffraction limited. Diffraction is always occurring when observing an object at any distance through any optics,

but it begins to dominate when the object's scale is equal or smaller than the Rayleigh criterion [4].

2.5.1.1 Rayleigh Criterion

The Rayleigh criterion states that light of wavelength λ will spread into a diffraction pattern with the first minimum of the distribution at an angular radius θ_R when passing through a circular aperture of diameter d [4] such that:

$$\begin{aligned}\sin \theta_R &= \frac{1.22\lambda}{d}, \\ \theta_R &\approx \frac{1.22\lambda}{d}.\end{aligned}\tag{2.62}$$

For a 0.5-meter aperture optical telescope with a pixel scale of 1 arcsecond per pixel, observing a 40-meter wide object in GEO — occupying approximately 0.2 pixel widths on the image — Eq 2.62 predicts that the diffraction pattern will be 2.4 times wider than the object at a wavelength of 550 nanometers. Similarly, a 60-centimeter wide object in LEO observed with the same telescope occupies 0.1 pixel widths in the image while producing an Airy disk 4.5 times wider than the object itself. As a result, GEO objects cannot be resolved from the ground, independent of atmospheric effects.

2.5.1.2 The Airy Disk

The far-field diffraction pattern produced by a point source is known as an Airy pattern or disk [4]. The Airy disk is expressed in terms of an amplitude C at an angular distance θ from the center $C(\theta)$ [4] as:

$$C_{\text{Airy}}(\theta) = C_0 \left(\frac{2J_1(k \cdot r_d \sin \theta)}{k \cdot r_d \sin \theta} \right).\tag{2.63}$$

In Eq 2.63, C_0 is the amplitude of the center of the Airy disk, r_d is the radius of the aperture, $k = \frac{2\pi}{\lambda}$ is the wavenumber, and J_1 is the first order Bessel function of the first kind. The central magnitude C_0 is expressed:

$$C_0 = \frac{S_{obj}^2 A_{aperture}^2}{2f^4}. \quad (2.64)$$

In Eq 2.64, S_{obj} is the total mean irradiance incident on the CCD due to the source, $A_{aperture}$ is the aperture area, and f is the focal length of the optics [4]. The pattern produced by Eq 2.63 is depicted in Figure 2.12.

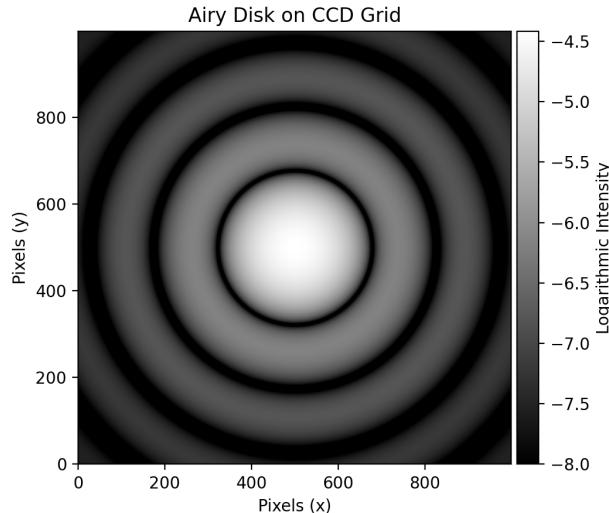


Figure 2.12. Airy disk diffraction pattern

The Rayleigh criterion expresses the angular size of the first zero of the Airy disk, after which the amplitude of the Airy disk drops off exponentially. It is often useful to approximate the Airy disk with a 2D Gaussian. Given the total signal to be approximated, this Gaussian is fit with a single parameter — the full width at half maximum (FWHM). The FWHM expresses the diameter at which the signal drops to half the magnitude of its central maximum [4]. The FWHM of the Airy disk is expressed [4]:

$$FWHM_{airy} = \frac{1.028\lambda}{2r_d}. \quad (2.65)$$

The diffraction pattern is not the only effect that spreads the unresolved signal over the pixel grid. Atmospheric turbulence contributes to further spreading and speckling of the signal [4]. This effect — known as the *seeing* — is encapsulated in $FWHM_{\text{seeing}}$ and is generally between 1 and 3 arcseconds [4]. While the seeing and diffraction pattern are additive, it is sufficient to take the larger value for simulation purposes [4]. The standard deviation of the Gaussian approximation of the Airy disk is given by:

$$\sigma = \frac{FWHM}{2\sqrt{2 \ln 2}}. \quad (2.66)$$

The full Gaussian approximation at an angular distance θ from the source is given by:

$$C_{\text{Gauss}}(\theta) = \frac{0.838\bar{C}_{\text{all}}}{2\pi\sigma^2} \exp\left(-\frac{\theta^2}{2\sigma^2}\right) \quad (2.67)$$

In practice, computing the Airy disk or its Gaussian approximation on rectangular pixel grid amounts to integrating the amplitude function $C(\theta)$ over the pixel area:

$$C_{\text{pix}}(x, y) = \int_x^{x+\Delta x} \int_y^{y+\Delta y} C(\theta(x, y)) dy dx. \quad (2.68)$$

The area within the first maximum of the Airy disk is given by [4]:

$$A_{\text{airy}} = \pi \left(\frac{648000/\pi}{s_{\text{pix}}} \sin^{-1} \left(\frac{1.22\lambda}{D} \right) \right)^2. \quad (2.69)$$

2.5.2 Signal-to-Noise Ratio (SNR)

Because the unresolved object signals are always superimposed on the background of the image, the SNR of a CCD is expressed with the signal included in the denominator noise term [4]:

$$SNR = \frac{S_{\text{obj}}}{\sqrt{S_{\text{obj}} + N}}. \quad (2.70)$$

Explicitly, the SNR can be expanded in terms of the signal means and variances, assuming that their respective probability density functions are independent [4]:

$$SNR = \frac{S_{obj}}{\sqrt{S_{obj+n_{pix}}(\lambda_{background} + \lambda_{dark} + \sigma_{read}^2 + \frac{g^2}{24})}}. \quad (2.71)$$

It should be noted that the SNR computed in Eq 2.71 is idealized. In reality, the discrete sampling of each random variable in each object pixel — as well as the Poisson sampling of the object signal — results in a slightly different value. As the number of object pixels approaches infinity, the SNR computed from the image would approach the theoretical value. Figure 2.13 illustrates the effect of raising the background noise level while holding the object signal constant on the SNR.

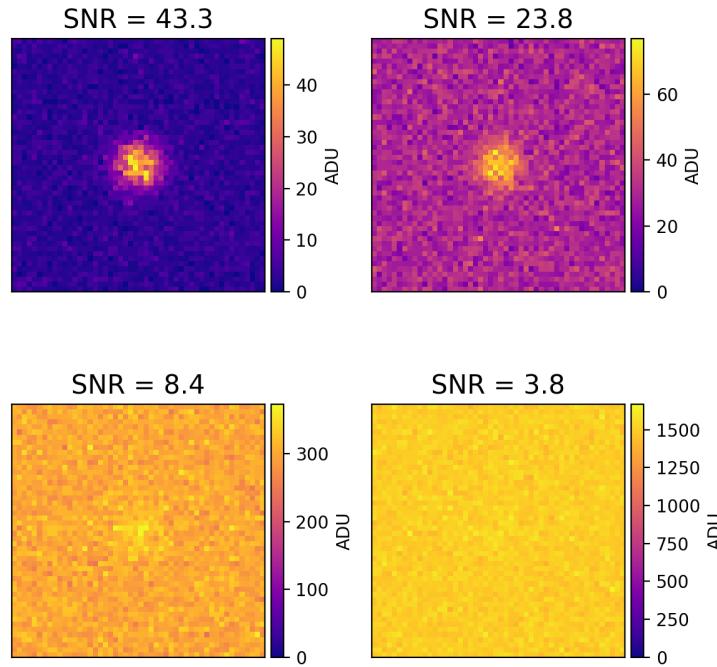


Figure 2.13. Signal-to-noise ratio reduction on a synthetic CCD pixel grid

The SNR is an important constraint when determining whether an object signal is visible above the background. If the SNR is too low, it becomes difficult or impossible to notice and quantify the object signal. Simulations in this work are all performed with a reasonable SNR limit of 3.

2.5.3 Telescope Properties

While much of the focus in this work is directed towards the CCD, the telescope it is mounted within has a few important properties which must be known for accurate CCD simulation.

The aperture of the telescope is the beginning of the optical path within the telescope — the area that collects incoming light. The area of the aperture $A_{\text{aperture}} = \pi r_d^2$, where r_d is the radius of the aperture must be distinguished from the *effective* aperture area, which is the total area of the aperture which actually collects light. Some telescopes have central obstructions which remove an amount of aperture area from the optical path, such that the effective aperture area is:

$$A_{\text{eff}} = A_{\text{aperture}} - \pi r_o^2, \quad (2.72)$$

where r_o is the radius of the obstruction. The telescope focal length f is the length of the optical path within the optics required to resolve a focused image. The pixel scale s_{pix} is a measure of the angle spanned by the side length of each square pixel, often expressed in arcseconds per pixel.

2.5.4 Brightness Units

In the context of photometry, "brightness" is a catch-all term for a variety of units. Defining the relationships between these units will make later light curve conversions more clear. Brightness conversions are needed for both light curves and the environmental data sources that drive the CCD performance model described in Section 2.5.

2.5.4.1 S_{10} Surface Brightness

While apparent magnitude and irradiance are effective for quantifying the flux of point sources, other units exist to describe diffuse or extended sources where brightness is spread over an area. S_{10} is a unit of surface brightness representing the number of 10th magnitude stars per square degree that would produce the same flux as a given diffuse source. Surface

brightness in S_{10} over a given solid angle Ω [sr] can be converted to total irradiance I $\left[\frac{W}{m^2}\right]$ via [9]:

$$\frac{I \left[\frac{W}{m^2}\right]}{S_{10}} = 10^{-10/2.5} \left(\Omega \frac{180^2}{\pi^2} \right) \int_{10^{-8}}^{10^{-6}} \text{STRINT}(\lambda) d\lambda = 8.26617 \Omega \cdot 10^{-9}. \quad (2.73)$$

In 2.73, $\text{STRINT}(\lambda) \left[\frac{W}{m^2 \cdot m}\right]$ is the representative spectrum of a 0th magnitude star, $\text{QE}(\lambda)$ is the quantum efficiency spectrum of the observing sensor, $\text{ATM}(\lambda)$ is the atmospheric transmission spectrum, λ [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. Quantum efficiency has units of photoelectrons, conveying the fraction of incident photons converted to photoelectrons in the CCD sensor. Atmospheric transmission is a unitless quantity conveying the fraction of light that is not absorbed by the atmosphere. Example spectra for $\text{ATM}(\lambda)$ and $\text{QE}(\lambda)$ are displayed in Figure 2.15, with underlying data provided in Appendix 7.2.0.1.

2.5.4.2 Magnitude per Square Arcsecond

A second surface brightness unit is $\left[\frac{mag}{arcsec^2}\right]$, also known as MPSAS (magnitude per square arcsecond) [9]. 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, the equivalent magnitude of a single point source is measured. A surface brightness B_{10} in S_{10} can be converted into surface brightness B_{mag} in $\left[\frac{mag}{arcsec^2}\right]$ by applying Eq 2.32.

$$B_{mag} = -2.5 \log_{10} \left(\frac{B_{10} \cdot 10^{-4}}{12960000} \right). \quad (2.74)$$

In Eq 2.74 B_{10} is converted to the total irradiance per square degree, converted square degrees to square arcseconds, and finally transformed into apparent magnitude. MPSAS is converted to irradiance per steradian using 2.32:

$$I = \left(\frac{180}{3600\pi} \right)^2 I_s \cdot 10^{-\frac{MPSAS}{2.5}}. \quad (2.75)$$

2.5.4.3 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" [65]. This means that an isotropic green light 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, candela B_{cd} is converted to watts per steradian B_{wsr} via [65]:

$$B_{wsr}(\lambda) = \frac{B_{cd}}{K_{cd}(\lambda)}. \quad (2.76)$$

The luminous efficiency function $K_{cd}(\lambda)$ models the human eye's response to the visible spectrum [66]. Different fits of this function exist; the function proposed Sharpe et al. is adopted, displayed in Figure 2.14 [66].

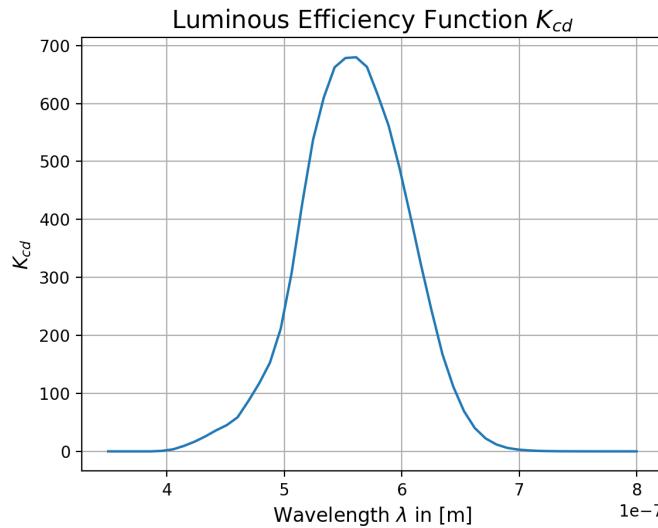


Figure 2.14. Luminous efficiency function from [66]

Candela per unit area can be converted into MPSAS by combining Eq 2.76 with 2.31, yielding a formulation which is still a function of the source's wavelength:

$$MPSAS(\lambda) = -2.5 \log_{10} \left(\frac{B_{cd}}{\left(\frac{180}{3600\pi} \right)^2 K_{cd}(\lambda) I_s} \right). \quad (2.77)$$

2.5.4.4 Photoelectron Counts

Accurately simulating measured light curves requires an accurate simulation of the CCD camera taking the image. The first step towards that is understanding how irradiance at the telescope aperture is converted into pixel values in the final image. Raw images taken by a CCD-equipped telescope have pixel values measured in photoelectron counts, otherwise known as Analog-to-Digital Units (ADU) [9]. The count in a single pixel obtained is directly proportional (via the CCD's gain) to the number of photons incident on that pixel during the integration time [9]. Higher order effects in the silicon of the CCD makes this description incomplete, but for non-resolved imaging applications concerned about, effects smaller than the sensor readout noise and dark current can be safely neglected [4]. Irradiance can be converted to ADU via the conversion factor $SINT$ through [9]:

$$SINT = A_{eff} \int_{10^{-8}}^{10^{-6}} \left(\frac{SUN(\lambda)}{I_{sun}} \right) \cdot QE(\lambda) \cdot ATM(\lambda) \cdot \left(\frac{\lambda}{hc} \right) d\lambda. \quad (2.78)$$

In Eq 2.78, $SUN(\lambda)$ 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 $\left[\frac{W}{m^2} \right]$, and A_{eff} is the effective aperture area. Read literally, the integral term has units $\left[\frac{1}{W_s} \right]$, giving the number of counts per incident Watt of solar radiation and second of integration time. The aperture diameter factor outside the integral accounts for the area of light incident on the CCD, giving SINT units of $\left[\frac{m^2}{W_s} \right]$. The spectra in Eq 2.78 are plotted in Figure 2.15 with data in Appendix 7.2. Multiplying by irradiance in $\left[\frac{W}{m^2} \right]$ and an integration time Δt in seconds will yield the mean photoelectron signal \bar{C}_{all} in ADU via:

$$\bar{C}_{all} = SINT \cdot I \cdot \Delta t. \quad (2.79)$$

For completeness, irradiance can be recovered from a signal in ADU and the integration time:

$$I = \frac{S}{\text{SINT} \cdot \Delta t}. \quad (2.80)$$

2.5.5 Background Signal Sources

2.5.5.1 Background Source Importance

Some background signals are more impactful than others. For simulating realistic light curves, it is important to only model those background sources that have the possibility of being at or above the order of magnitude of the object signal, thereby seriously degrading the signal-to-noise ratio. As a baseline, the background terms modeled by Krag for the PROOF CCD performance model were implemented, namely scattered moonlight, airglow, zodiacal light, and integrated starlight [9]. In addition, twilight and light pollution models were implemented to increase the time and station location flexibility of the simulation. Each background term is modeled at medium fidelity — often using tabulated measurements or a set of exponential distributions to govern the scattering physics. As a result, this background performance model does not capture local weather conditions or precisely simulate the scattering of each photon through the atmosphere, but still strives to faithfully model the physics of each background signal process. Table 2.3 ranks the approximate magnitudes in photoelectrons per pixel one can expect from a telescope similar to the Purdue Optical Ground Station.

2.5.5.2 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. Each spectrum is displayed in Figure 2.15.

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.

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

Table 2.3. Background signal importance, values generated by the author using models by Krag and Daniels [9], [67]

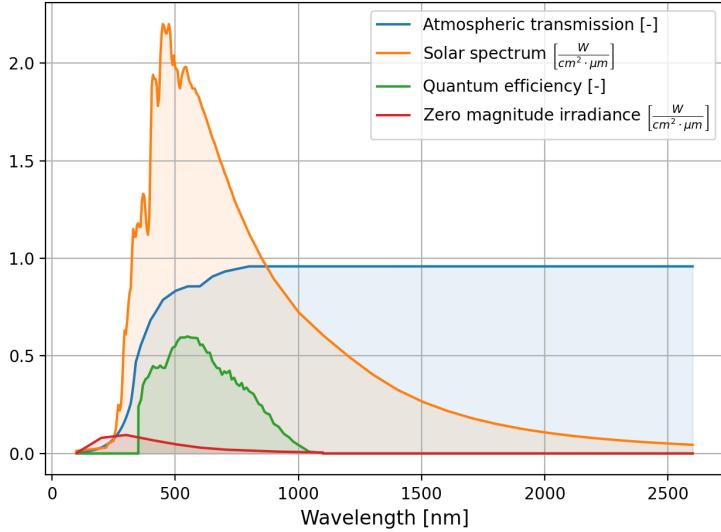


Figure 2.15. Astronomical Spectra, atmospheric transmission and zero magnitude stellar spectrum from [9]

2.5.5.3 Airglow

Certain chemical reactions from 80-110 km altitude in the upper atmosphere release visible light [9]. 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. The airglow signal AINT is modeled in a similar fashion to integrated starlight. Given the airglow spectra $\text{GLINT}(\lambda) \left[\frac{W}{m^2 \cdot sr} \right]$, the sensor quantum efficiency $\text{QE}(\lambda)$, the atmospheric transmission $\text{ATM}(\lambda)$, as well as Plank's constant h and the speed of light in vacuum c , the airglow signal is computed via [9]:

$$\text{AINT} = A_{\text{aperture}} \int_{10^{-8}}^{10^{-6}} \text{GLINT}(\lambda) \cdot \text{QE}(\lambda) \cdot \text{ATM}(\lambda) \cdot \left(\frac{\lambda}{hc} \right) d\lambda. \quad (2.81)$$

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

$$\bar{S}_{\text{airglow}} = \text{AINT} \cdot \text{AM}(\theta_z) \cdot \Delta t \cdot \left(\frac{\pi s_{\text{pix}}}{648000} \right)^2. \quad (2.82)$$

In Eq 2.82, $AM(\theta_z)$ is the relative airmass function which accounts for the accumulation of air along the optical path at different zenith angles [4]. This airmass is termed *relative* as it relates the ratio of absolute airmass at a zenith angle to the absolute airmass at zenith. Often, this function is approximated by the Van-Rhijn factor $AM(\theta_z) = \sec \theta_z$, which remains accurate up to $\theta_z \approx 70^\circ$ before diverging to infinity. Instead, a function proposed by Pickering is used [68].

$$AM(\theta_z) = \frac{1}{\sin \left((90 - \theta_z) + \frac{244}{165+47*(90-\theta_z)^{1.1}} \right)} \quad (2.83)$$

Using Eq 2.83 instead of the Van-Rhijn factor enables the computation of background signals near the horizon without the singularity of $\sec \theta$. Figure 2.16 displays this comparison in action.



Figure 2.16. Airmass function comparison. The Van-Rhijn factor diverges to $+\infty$ while Pickering's function reaches the correct maximum of $AM(\theta_z) \approx 40$.

2.5.5.4 Light Pollution

Another source of background noise is light pollution. On a cloudless night with low levels of atmospheric aerosols, the zenith surface brightness is approximately $22 \left[\frac{\text{mag}}{\text{arcsec}^2} \right]$ (MPSAS)



Figure 2.17. Mean airglow signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W

[9]. 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, the 2015 World Atlas of Sky Brightness dataset is used [69]. The data is reported in $\left[\frac{mag}{cm^2} \right]$ on a 30-arcsecond grid, requiring conversion to a more useful unit. A subset of the global dataset is displayed in 2.18 This conversion is listed in Eq 2.77, using a monochromatic $\lambda = 474$ nm to fit the conversions of Falchi et al. [70].

The mean light pollution CCD signal in ADU per pixel is formulated similarly to airglow. The station's zenith surface brightness $B_{poll,z}$ in MPSAS, linearly interpolated from the World Atlas dataset, is converted to irradiance per steradian via 2.75 and to ADU per pixel via:

$$\bar{S}_{pollution} = B_{poll,z} \cdot SINT \cdot AM(\theta_z) \cdot \Delta t \cdot \left(\frac{\pi s_{pix}}{648000} \right)^2. \quad (2.84)$$

Note that Krag does not implement a specific light pollution model, but instead takes the dark sky site zenith brightness of 22 MPSAS as input to an atmospherically scattered light model [9]. The formulation of the light pollution model in Eq 2.84 is simply an adaptation of Krag's model with a variable zenith brightness driven by a light pollution dataset.

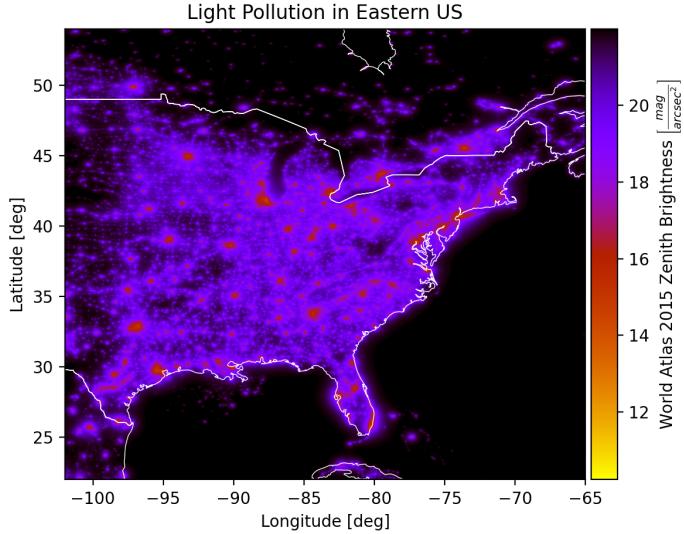


Figure 2.18. Zenith light pollution in the eastern USA, data from [69]

2.5.5.5 Twilight

Even after the Sun sets, scattered sunlight in the upper atmosphere creates a signal on the 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 [71]. 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. The model of Patat et al. fits a second-degree polynomial in γ to approximately 2000 observations in varying atmospheric conditions, yielding separate curves for each of the UBVRI passbands. For example, for the V band, the twilight zenith brightness in MPSAS is given by [71]:

$$B_{twi,z} = 11.84 + 1.518(\gamma - 95^\circ) - 0.057(\gamma - 95^\circ)^2. \quad (2.85)$$

Eq 2.85 is valid from $95^\circ \leq \gamma \leq 105^\circ$. While $\gamma \leq 95^\circ$, the zenith brightness is taken to be constant and equal to the brightness at $\gamma = 95^\circ$. This is not accurate, as it predicts daylight to be the brightness of twilight, but is sufficiently bright to correctly forbid daytime observations by lowering the SNR drastically. After $\gamma = 105^\circ$ the zenith surface brightness is



Figure 2.19. Mean light pollution signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W

set to $B_{twi,z} = 22$ MPSAS to match the optimal observation condition of the light pollution model [9]. Zenith twilight brightness is plotted as a function of γ in Figure 2.20.

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

$$\bar{S}_{twilight} = B_{twi,z} \cdot SINT \cdot AM(\theta_z) \cdot \Delta t \cdot \left(\frac{\pi s_{pix}}{648000} \right)^2 \quad (2.86)$$

2.5.5.6 Integrated Starlight

Stars are almost always present in optical images of space objects. The brightest stars streaking across the field of view in Figure 2.22 have high SNRs and stand out clearly against the dark background. This raises a question: if the telescope observes a full $1^{\circ} \times 1^{\circ}$ area of the sky, where are the rest of the stars? The Milky Way alone contains approximately $1 \cdot 10^{11}$ stars. The answer is clear: many more stars are present in the image, most of them falling into the background. This residual faint starlight is termed "integrated" starlight. Computing



Figure 2.20. Twilight model surface brightness at zenith as a function of solar zenith angle

the brightness of the integrated starlight in the background of the image circumvents the need to model all the stars in the catalog.

Krag [9] modeled this signal by building a $1^\circ \times 1^\circ$ grid of surface brightness values for the full inertial sphere, parameterized by RA/Dec. Krag used the Guide Star catalog, which contains 15 million stars down to apparent magnitude 16. Exponential extrapolation was used to predict star counts in each bin for higher magnitudes [9]. Twenty years later, larger star catalogs exist that are nearly complete to much higher apparent 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 [72]. The same $1^\circ \times 1^\circ$ grid was computed using GAIA [73], resulting in Figure 2.23 which shows the computed brightness map in units of S_{10} .

With this map of exoatmospheric mean brightness of the night sky due to integrated starlight, the corresponding signal mean in the telescope CCD is computed, adopting Krag's formulation [9].



Figure 2.21. Mean twilight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W

$$\text{BINT} = A_{\text{aperture}} \int_{10^{-8}}^{10^{-6}} \text{STRINT}(\lambda) \cdot \text{QE}(\lambda) \cdot \text{ATM}(\lambda) \cdot \left(\frac{\lambda}{hc} \right) d\lambda \quad (2.87)$$

In Eq 2.87, 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.

$$\bar{S}_{\text{star}} = 10^{-4} \cdot \text{BINT} \cdot \left(\frac{s_{\text{pix}}}{3600} \right)^2 \cdot \Delta t \cdot b_{\text{is}} \quad (2.88)$$

In Eq 2.88, b_{is} is the integrated starlight brightness in $[S_{10}]$ computed by linearly interpolating the dataset in Figure 2.23, s_{pix} is the telescope pixel scale in $\left[\frac{\text{arcsecond}}{\text{pix}} \right]$ 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 $\left[\frac{e^-}{\text{pix}^2} \right]$; photoelectron counts (ADU) per pixel. Figure 2.24 shows the background signal mean due to integrated starlight.



Figure 2.22. Raw image of three GEO objects with stars streaking through the background. As expected the star signals have a variety of signal-to-noise ratios. Taken by the Purdue Optical Ground station at 32.900° N, -105.533° W by Nathan Houtz.

2.5.5.7 Scattered Moonlight

Moonlight scattering through the atmosphere significantly increases background brightness [9]. 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 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 [67]. Interpolating this table yields the intensity of the Rayleigh scattering F_{rs} in $10^{-10} \text{ W}/(\text{cm}^2 \cdot \mu\text{m} \cdot \text{sr})$ [9]. The Mie scattered component is formulated [9]:

$$F_{ms}(\lambda) = a_1 \left[e^{-\left(\frac{\Psi}{\Psi_1}\right)} + a_2 e^{-\left(\frac{\pi-\Psi}{\Psi_2}\right)} \right] F_{rs}(\lambda). \quad (2.89)$$

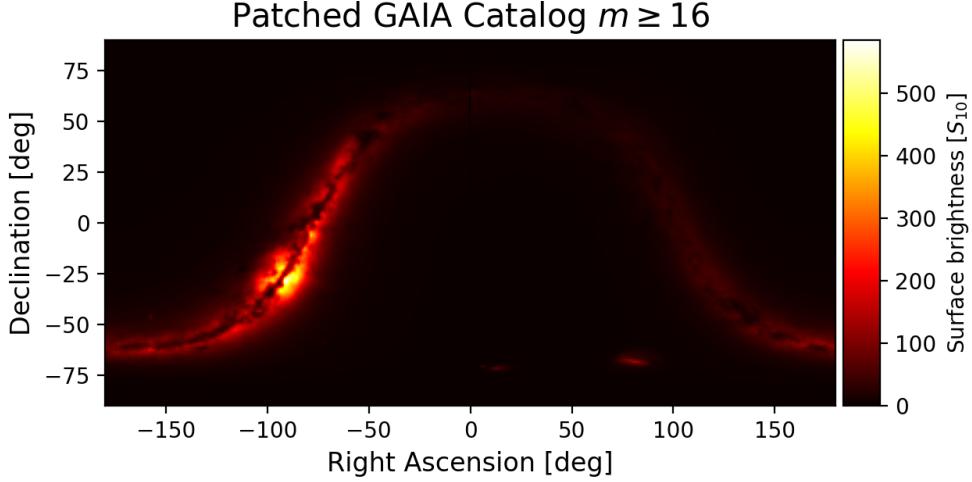


Figure 2.23. Integrated starlight brightness map

Daniels recommends $a_1 \in [50, 100]$, $a_2 \in [0.01, 0.02]$, $\Psi_1 \in [10^\circ, 20^\circ]$, and $\Psi_2 \approx 50$ [67]. Prior to any station-specific fitting, the center of each interval is chosen, yielding $a_1 = 75$, $a_2 = 0.015$, $\Psi_1 = 15^\circ$, and $\Psi_2 = 50^\circ$. a_1 and a_2 are dimensionless, such that F_{ms} also has units of $10^{-10} W/(cm^2 \cdot \mu m \cdot sr)$. The total intensity of the scattered moonlight F_{mt} following Krag's formulation [9]:

$$F_{mt} = f(\theta) [F_{rs}(\lambda) + F_{ms}(\lambda)] . \quad (2.90)$$

in Eq 2.90, $f(\theta)$ is the lunar phase function which describes the fraction of the full Moon brightness reflected at an observer when the Sun-Moon-observer angle is θ . This function is linearly interpolated within Table 3 in [67]. 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 to be [9]:



Figure 2.24. Integrated starlight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W

$$f_{corr} = \frac{I_s}{SUN(550 \text{ [nm]})}. \quad (2.91)$$

With all these pieces, the mean scattered moonlight signal in ADU per pixel is [9]:

$$\bar{S}_{moon} = F_{mt}(550 \text{ [nm]}) \cdot SINT \cdot \left(\frac{s_{pix}}{3600} \right)^2 \cdot \Delta t \cdot f_{corr}. \quad (2.92)$$

2.5.5.8 Zodiacal Light

Zodiacal light is an effect created by sunlight reflecting off of dust in the ecliptic plane [9]. Zodiacal light is strongest around the Sun — an exclusion zone for most optical telescopes — 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". The zodiacal light brightness is linearly interpolated within Table 1 of [74] which is listed for convenience in Appendix 7.2.3. This reports the surface brightness of the zodiacal light in S_{10} , which is used without conversion to find the mean CCD signal in ADU per pixel via:



Figure 2.25. Mean scattered moonlight signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W

$$\bar{S}_{zod} = BINT \cdot \left(\frac{s_{pix}}{3600} \right)^2 \cdot \Delta t \cdot ZOD \cdot 10^{-4}. \quad (2.93)$$

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

2.5.5.9 Background Sampling

The background signals are only defined in terms of their means, as each signal models the expected amount of radiation without accounting for the quantized nature of light [9]. Since light is transmitted in individual photons, their incidence on a given pixel will follow a statistical distribution. Assuming that each photon does not interact with others, the incidence of a photon on a pixel is well modeled as a Poisson process for each background term [4]. This distribution models the number of independent and identically distributed events that occur during a time period. For CCD astronomy, this translates to the event of a photon hitting the sensor. A Poisson distribution is defined on the positive integers by a

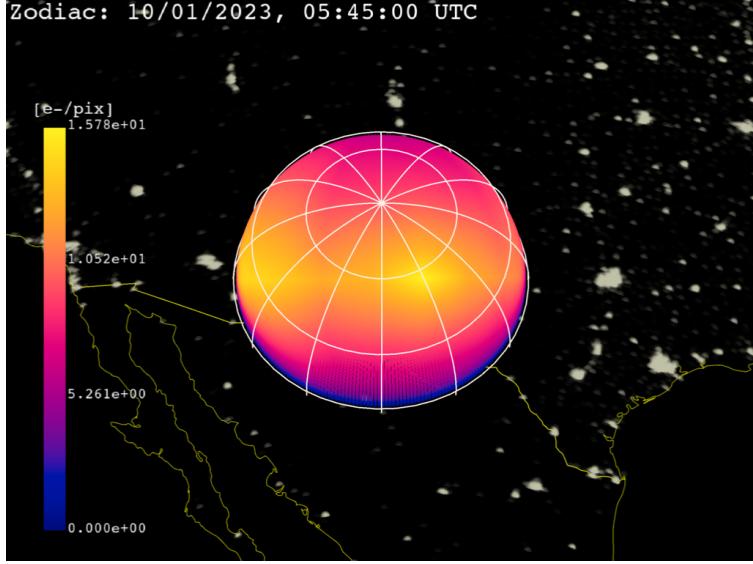


Figure 2.26. Mean zodiacal light signal on the local observer hemisphere. The observer is in New Mexico, USA at 32.900° N, -105.533° W

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 [4]:

$$P_\lambda(x = k) = \frac{\lambda^k e^{-\lambda}}{k!}. \quad (2.94)$$

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. The background sources modeled in this work are reasonably assumed to be independent as they each originate from distinct physical processes.

$$\lambda_{background} = \bar{S}_{airglow} + \bar{S}_{pollution} + \bar{S}_{twilight} + \bar{S}_{star} + \bar{S}_{moon} + \bar{S}_{zod} \quad (2.95)$$

Drawing samples from the Poisson distribution defined by $\lambda_{background}$ computes the background of the CCD image.

2.5.6 Sensor Effects

CCD sensors accumulate noise during both the integration time and readout phases. Characterizing these noise sources is necessary to sample realistic light curves and compute the signal-to-noise ratio of the measurements.

2.5.6.1 Dark Noise

The dark noise, also called the dark current or dark count, captures the temperature-dependent accumulation of electrons in the CCD pixel wells [9]. This noise source is modeled as a Poisson process with parameter λ_{dark} [4] and is assumed to be independent from the other sensor effects. This source accumulates with the integration time, giving it units of counts per second [9].

2.5.6.2 Readout Noise

When the CCD is read out, the charge contained in each pixel well must be digitized. This process introduces noise in the final signal due to electronic effects within the CCD circuitry and its surrounding environment [9]. The readout noise is modeled as a zero-mean Gaussian distribution with variance σ_{read}^2 and is also assumed to be independent from other sensor effects [4].

2.5.6.3 Truncation Noise

Truncation noise in a CCD stems from the fact that the charge in each pixel is digitized into an integer factor of the gain [4]. This is modeled using a uniform distribution on $[-g/2, g/2]$, yielding a variance $N_{trunc}^2 = \frac{g^2}{24}$ [4].

The particular noise variances for the Purdue Optical Ground Station are listed in 7.1.

2.5.7 Observer Constraints

Accurately simulating light curves requires realistic constraints on the observing station. For an optical ground station, basic operational constraints include local eclipse, target

SNR, Moon exclusion angle, and minimum elevation constraints. This section develops these constraints.

2.5.7.1 Earth Penumbra

The Earth casts a conical shadow into space. Outside this shadow, the fraction of received solar irradiance f_{Sun} is assumed to be 1. Inside the shadow, the central region of zero received solar irradiance $f_{Sun} = 0$ is known as umbra, while the boundary between umbra and full Sun is known as the penumbra where $f_{Sun} \in (0, 1)$. The precise gradient of the penumbra is defined by the geometry of the overlapping Earth and Sun disks from the perspective of the object. Given the position of the Sun \mathbf{r}_{Sun} computed following Section 2.4.8 and object in J2000, the local irradiance fraction a function of the angles [9]:

$$\begin{aligned} \mathbf{r}_{obj2Sun} &= \mathbf{r}_{Sun} - \mathbf{r}_{obj} & (2.96) \\ \epsilon_s &= \sin^{-1} \left(\frac{R_{Sun}}{\|\mathbf{r}_{obj2Sun}\|} \right) \\ \epsilon_e &= \sin^{-1} \left(\frac{R_{Earth}(\phi_{obj})}{\|\mathbf{r}_{obj}\|} \right) \\ \epsilon &= \cos^{-1} (-\hat{\mathbf{r}}_{obj2Sun} \cdot \hat{\mathbf{r}}_{obj}) \\ \epsilon_1 &= \frac{\epsilon^2 - \epsilon_e^2 + \epsilon_s^2}{2\epsilon} \\ \epsilon_2 &= \frac{\epsilon^2 + \epsilon_e^2 - \epsilon_s^2}{2\epsilon}. \end{aligned}$$

In Eq 2.96, ϵ_s is the half-angle of the Sun disk at the object, ϵ_e is the half-angle of the Earth disk from the object as a function of the geodetic latitude of the object ϕ_{obj} . Additionally, ϵ is the angle between the center of the Earth and Sun from the perspective of the object, with ϵ_1 and ϵ_2 being the angles from the center of the Sun and Earth disks to the center of their overlap, if it exists [9]. These angles are defined in Figure 2.27. Given these angles, the illumination fraction f_{Sun} is computed:



Figure 2.27. Angle and area definitions for penumbra shadowing, Figure 5.20 in [9]

$$f_{Sun} = \begin{cases} 1 & \epsilon \geq (\epsilon_s + \epsilon_e) \\ 0 & \epsilon \leq (\epsilon_e - \epsilon_s) \\ \frac{a_1 - a_2}{\pi \epsilon_s^2} & \text{else} \end{cases} \quad (2.97)$$

In Eq 2.97, the solid angles a_1 and a_2 are defined in Figure 2.27 and are computed via:

$$\begin{aligned} a_1 &= \pi \epsilon_s^2 - \epsilon_s^2 \cos^{-1} \left(\frac{\epsilon_1}{\epsilon_s} \right) + \epsilon_1 \sqrt{\epsilon_s^2 - \epsilon_1^2} \\ a_2 &= \epsilon_e^2 \cos^{-1} \left(\frac{\epsilon_2}{\epsilon_e} \right) - \epsilon_2 \sqrt{\epsilon_e^2 - \epsilon_2^2} \end{aligned} \quad (2.98)$$

Computing Eq 2.97 for a dense grid of points in the J2000 reference frame reveals the geometry of the penumbra and umbra in Figure 2.28, illustrating that the penumbra expands into the umbra as the observer recedes from the Earth.

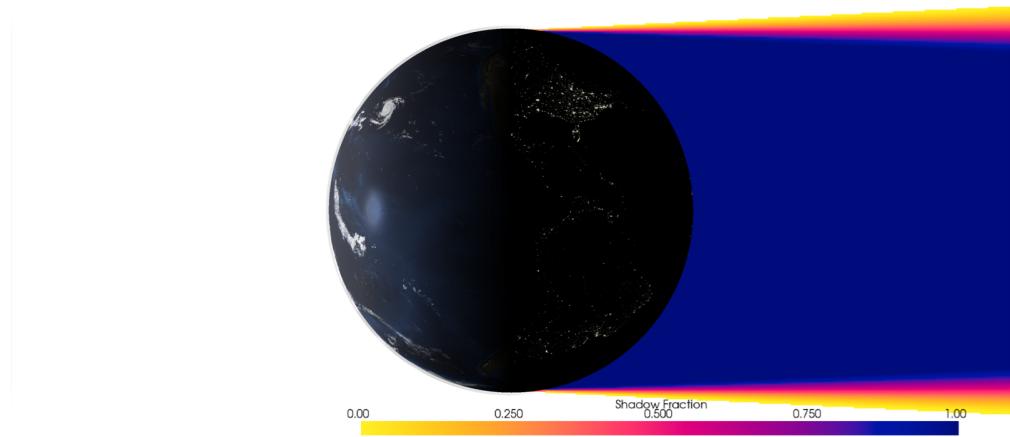


Figure 2.28. Shadow fraction in J2000, effect exaggerated by scaling the Sun radius by a factor 10

2.5.7.2 SNR

As the signal-to-noise ratio of an object signal on the CCD pixel grid degrades, it becomes more difficult to determine whether an object is present at all. Given a minimum limiting SNR SNR_{min} this constraint takes the form:

$$\nu_{\text{SNR}} = \begin{cases} 1 & \text{SNR} > \text{SNR}_{min} \\ 0 & \text{else} \end{cases}. \quad (2.99)$$

In Eq 2.100, $\nu_{\text{SNR}} = 1$ implies that the observation is valid, with $\nu_{\text{SNR}} = 0$ implying that the object signal cannot be discerned from the image background.

2.5.7.3 Moon Exclusion Angle

Focusing the irradiance of the Moon onto the sensor grid will seriously damage the CCD. For the Purdue Optical Ground Station, straylight from the Moon begins leaking into the optics around 30° from the center of the Moon disk. This observation constraint is imposed for an arbitrary exclusion angle $\theta_{\text{Moon},ex}$ via:

$$\nu_{Moon} = \begin{cases} 1 & \cos^{-1} \left(\frac{(\mathbf{r}_{Moon} - \mathbf{r}_{obs}) \cdot (\mathbf{r}_{obj} - \mathbf{r}_{obs})}{\|\mathbf{r}_{Moon} - \mathbf{r}_{obs}\| \|\mathbf{r}_{obj} - \mathbf{r}_{obs}\|} \right) > \theta_{Moon,ex} \\ 0 & \text{else} \end{cases}. \quad (2.100)$$

In Eq 2.100, $\nu_{Moon} = 1$ implies that the observation is valid, with $\nu_{Moon} = 0$ implying that the observation is too close to the Moon. \mathbf{r}_{Moon} should be computed with SPICE following Section 2.4.8.

2.5.7.4 Minimum Elevation

Due to the local topology at the observer location, an operator may assign a minimum angle from the horizon $\theta_{elev,min}$ for observations. Approximating the station inertial position \mathbf{r}_{obs} as being aligned with zenith, this yields the condition:

$$\nu_{elevation} = \begin{cases} 1 & 90^\circ - \cos^{-1} \left(\frac{(\mathbf{r}_{obj} - \mathbf{r}_{obs}) \cdot (\mathbf{r}_{obs})}{\|\mathbf{r}_{Moon} - \mathbf{r}_{obs}\| \|\mathbf{r}_{obs}\|} \right) > \theta_{Moon,ex} \\ 0 & \text{else} \end{cases}. \quad (2.101)$$

In Eq 2.100, $\nu_{elevation} = 1$ implies that the observation is valid, with $\nu_{elevation} = 0$ implying that the observation is too close to the horizon. \mathbf{r}_{Moon} is be computed using SPICE following Section 2.4.8.

2.5.8 Mean Irradiance at The Observer

Given the normalized irradiance computed via Eq 2.35 for convex objects or Eq 2.43 for any object shaded using Algorithm 4, the received mean irradiance at the telescope aperture is given by:

$$\bar{I} = \nu_{Moon,ex} \cdot \nu_{elevation} \cdot \nu_{SNR} \frac{I_s f_{Sun} \hat{I}}{(\mathbf{r}_{obj} - \mathbf{r}_{obs})^2}. \quad (2.102)$$

In Eq 2.102, the conditions $\nu_{Moon,ex}$, $\nu_{elevation}$, ν_{SNR} determine whether the telescope can take the observation and identify the object in the image, $I_s f_{Sun}$ determines the solar irradiance incident on the object, and the remaining terms $\hat{I}/(\mathbf{r}_{obj} - \mathbf{r}_{obs})^2$ dimensionalize

the normalized irradiance. In order to sample a noisy light curve, this mean irradiance must be converted into a mean photoelectron count via Eq 2.79, yielding \bar{C}_{all} :

$$\bar{C}_{all} = \text{SINT} \cdot \bar{I} \cdot \Delta t. \quad (2.103)$$

2.5.9 Sampling Noisy Light Curves

Given the irradiance of the object observed by the telescope, the noisy light curve is computed by building a grid containing the object signal, background noise, and sensor noise. On a pixel-by-pixel basis, the mean object signal is given by an alteration of Eq 2.67 where the angle θ from the center of the frame is converted into pixels via $\theta^2 = ((x - x_0)^2 + (y - y_0)^2) / s_{pix}^2$:

$$\bar{C}_{obj}(x, y) = \frac{0.838\bar{C}_{all}}{2\pi\sigma^2} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2 s_{pix}^2}\right). \quad (2.104)$$

In Eq 2.104, (x_0, y_0) are the exact pixel coordinates of the object centroid, σ is the Gaussian standard deviation from Eq 2.66 in arcseconds, and s_{pix} is the pixel scale in arcseconds per pixel. Likewise, the total noise sampled in each pixel is given by samples from all the relevant source distributions:

$$C_{noise} = N_{background} + N_{dark} + N_{trunc} + N_{read}. \quad (2.105)$$

In Eq 2.105, $N_{background}$ is a sample drawn from $\text{Pois}(\lambda_{background})$, N_{dark} is drawn from $\text{Pois}(\Delta t \cdot \lambda_{dark})$, N_{trunc} is drawn from $\text{Uniform}(-g/2, g/2)$, and N_{read} is drawn from $\text{Normal}(0, \sigma_{read}^2)$.

Assuming perfect knowledge of the image background mean in the region around the object signal — a reasonable assumption given that there are generally millions of background pixels in the image — the counts $C_{obj,meas}$ attributed to the object pixels $\{x_{obj}, y_{obj}\}$ is given by:

$$C_{noise,all} = \sum_{i=1}^{A_{airy}} C_{noise,i} \quad (2.106)$$

$$C_{obj,all} = \sum_{x \in x_{obj}, y \in x_{obj}} \bar{C}_{obj}(x, y)$$

$$C_{obj,meas} = C_{noise,all} - A_{airy} (\Delta t \cdot \lambda_{dark} + \lambda_{background})$$

In Eq 2.106, A_{airy} is the area of the central maximum of the Airy disk in square pixels presented in Eq 2.69. The measured object signal $C_{obj,meas}$ can then be transformed into irradiance via Eq 2.80:

$$I_{obj,meas} = \frac{C_{obj,meas}}{\text{SINT} \cdot \Delta t} \quad (2.107)$$

or to apparent magnitude by applying Eq 2.32:

$$m_{obj,meas} = -2.5 \log_{10} \left(\frac{I_{obj,meas}}{I_0} \right) \quad (2.108)$$

or normalized irradiance by applying Eq 2.33:

$$\hat{I}_{obj,meas} = \frac{\|\mathbf{r}_{obj} - \mathbf{r}_{obs}\|^2}{I_s} I_{obj,meas}. \quad (2.109)$$

2.5.10 Light Curve Simulation Results

2.5.10.1 Case 1: Normalized Light Curve for a Cube

This section presents the entire process for simulating the normalized light curve of a simple object. To begin, the properties of the object and the observing station are defined in Table 2.4.

To begin, the position of the object in TEME is computed using SGP4. That position is converted to ICRF using Eqs 2.57, 2.58, and 2.59. The position of the station in ITRF is computed using Eq 2.52 and converted to ICRF using Eqs 2.55, 2.56, 2.57, 2.58, and 2.59. The position of the Sun is determined with SPICE following Section 2.4.8.

Parameter	Value
Object	<code>cube.obj</code> 7.4
q_0	$[0, 0, 0, 1]^T$
w_0	$[0.04082483, 0.08164966, 0.04082483]^T$
J	$\text{diag}(1.0, 2.0, 3.0) [kg \cdot m^2]$
BRDF	Phong
C_s	0.3
C_d	0.7
n	5
Initial date	2023-03-26 05:00:00 UTC
Δt	0.1 [s]
n_{obs}	360 [s]
ϕ_{geod}	32.900°
λ	-105.533°

Table 2.4. Object and station simulation parameters

The attitude profile of the object is propagated with Eqs 2.10 and 2.9 using $\mathbf{M} = \mathbf{0}$ for torque free rigid body motion. The resulting attitude time history produces $q(t)$, $\omega(t)$, as well as the body frame vectors from the object to the Sun and observer depicted in Figure 2.29.

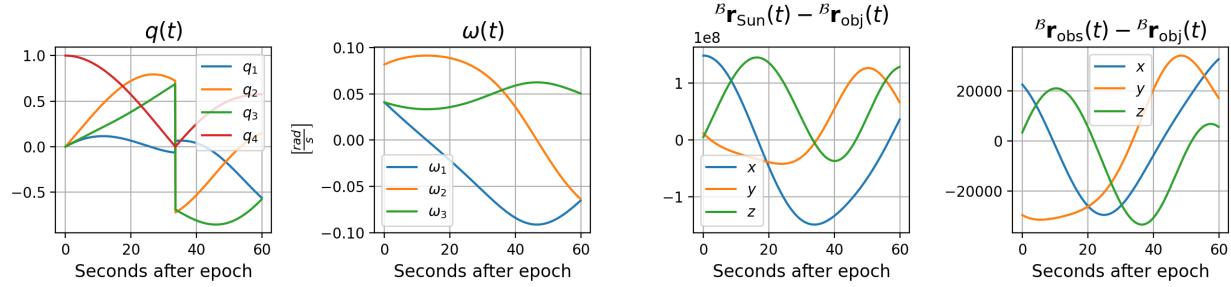


Figure 2.29. Attitude time history for case 1

Next, the BRDF is used to compute the reflection matrix for the cube with Eq 2.113. The normalized light curve is simply computed with Eq 2.35, using Eq 2.20 and noting that $\mu_{ij} = 0$ as the object is convex. This yields the normalized light curve depicted in Figure 2.30.

2.5.10.2 Realistic Light Curve for a Box-Wing Satellite

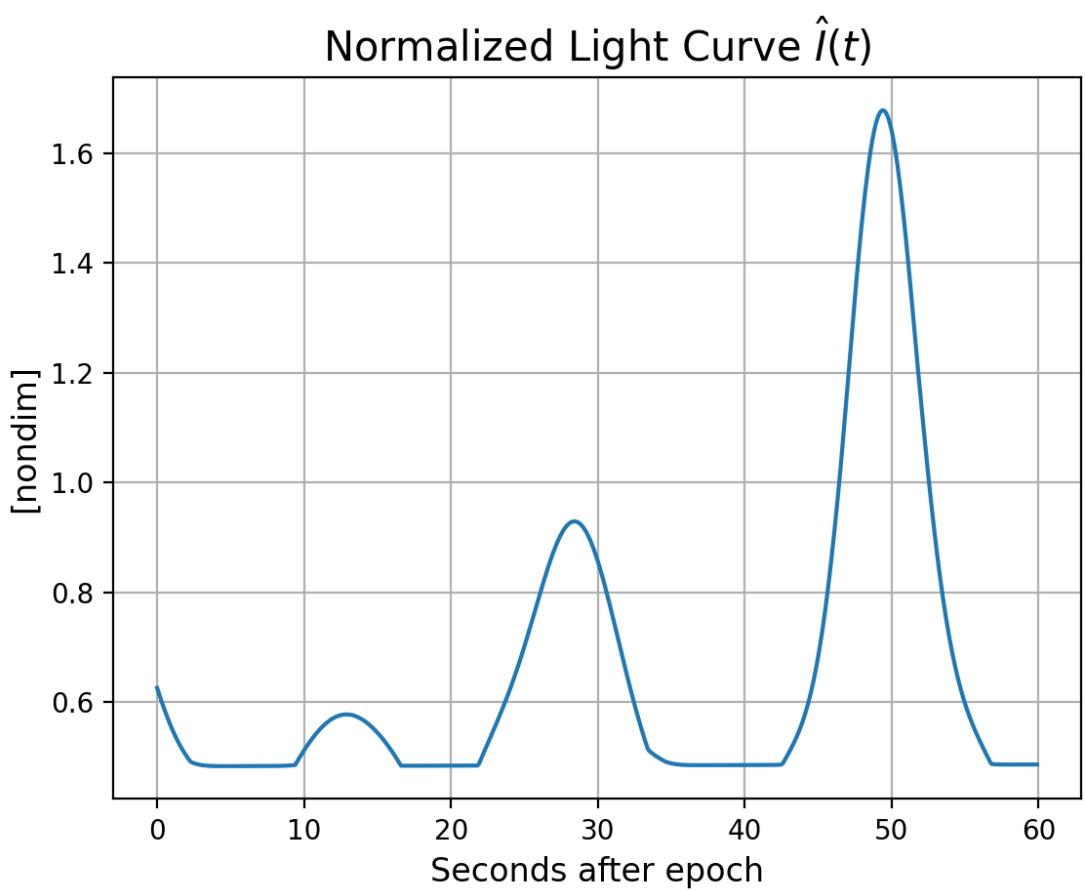


Figure 2.30. Normalized light curve for case 1

2.6 Light Curve Shape Inversion

2.6.1 Direct Convex Shape Inversion

Traditionally, direct light curve inversion involves two distinct optimization problems: a linear least squares problem to optimize normal vectors and areas to match the measured light curve, and a second optimization to produce accurate vertex positions and face adjacency information [5], [16], [30], [75]. The first problem is data-driven and linear, using the observations to estimate a plausible EGI. The second problem is highly nonlinear but convex and requires significant tuning for robust convergence [38]. The full convex inversion process is outlined:

1. Area and normal vector optimization
 - (a) Initial normal vector sampling
 - (b) Initial face area optimization
 - (c) Normal vector resampling TODO: should I note which steps are new contributions?
 - (d) Face area reoptimization
 - (e) Face merging
2. Object reconstruction
 - (a) Face support optimization
 - (b) Dual transform yielding final vertex locations
 - (c) Convex hull yielding final face adjacecies

2.6.1.1 The Extended Gaussian Image

In the continuous case, the Gaussian Image associates each point on the surface of a shape with a outward-pointing normal vector direction on the unit sphere \mathbb{S}^2 [76]. The determinant of the Jacobian of this forward mapping yields the Gaussian curvature $\kappa(u, v)$ of the surface

at each surface point — expressing how rapidly local normal vectors change direction [76]. The continuous *Extended Gaussian Image* (EGI) $E(\theta, \phi)$ is a function on the unit sphere parameterized by an azimuth θ and elevation ϕ which sums the inverse Gaussian curvature for all points (u_i, v_i) on the surface of the object with a normal pointing towards (θ, ϕ) [76]:

$$\text{??}E(\theta, \phi) = \sum_i \frac{1}{\|\kappa(u_i, v_i)\|}. \quad (2.110)$$

By Eq ??, flat areas of the object are mapped to point masses in the EGI, motivating a simpler definition of the EGI for discrete surfaces. The discrete EGI $\vec{E} \in \mathbb{R}^{m \times 3}$ is composed of m unit vectors \hat{n} each scaled by a nonnegative areaa $a \in \mathbb{R}$, where $a_i \geq 0 \forall i$ [32].

$$\vec{E}_i = a_i \hat{n}_i \quad (2.111)$$

In the context of shape inversion, the m vectors \hat{n} should be a relatively uniform tessellation of 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 $\vec{a} \in \mathbb{R}^{m \times 1}$. The norm of the EGI is notated $\|\vec{E}\| = \vec{a}$ 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 any EGI satisfying the closure condition [31]:

$$\sum_{i=1}^m a_i \hat{n}_i = [0, 0, 0]. \quad (2.112)$$

Equivalently, an EGI uniquely represents a closed, convex polyhedron — a polytope — with no open boundaries, up to a translation. While a given EGI uniquely represents a polytope, that same EGI could also be interpreted to be an infinite number of nonconvex and open geometries. An example of this extended family is depicted in Figure 2.31.

2.6.1.2 EGI Optimization

The EGI fulfills two important criteria for the shape inversion problem: it can be estimated directly from the light curve, attitude profile, and material properties, and uniquely

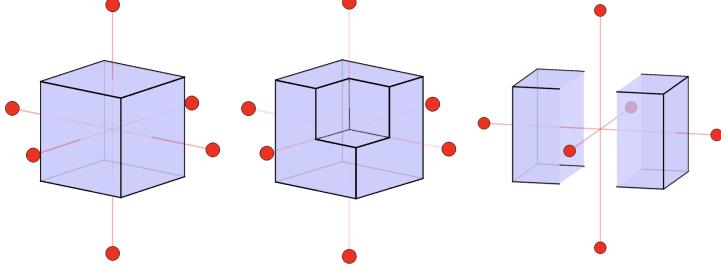


Figure 2.31. Simplified convex, nonconvex, and open EGI nonuniqueness. Larger circles indicate greater relative areas assigned to a given normal vector.

represents a convex object [10]. Further, the EGI can be transformed into a unique convex object and vice versa through the dual transform and Minkowski problem [31], [77].

To accomplish the goal of fitting a convex object to a collection of light curve measurements, it is advantageous to define a compact expression for the expected normalized irradiances of a convex object. This is accomplished through the convex reflection matrix $G \in \mathbb{R}^{n \times m}$ with ijth entries G_{ij} defined at time i for each facet j is defined using Eq 2.16 and Eq 2.35 as the normalized received facet irradiance per unit facet area:

$$G_{ij} = \frac{I_{ij}}{I_s a_j} \quad (2.113)$$

$$= f_r(L, O) (L_j \cdot N_i) (O_j \cdot N_i).$$

It must be noted that Eq 2.113 assumes that the BRDF of the object is uniform across its surface. This is a core assumption of the direct convex inversion scheme in the literature, and a limitation of the method that cannot be easily circumvented. This relationship between the received normalized irradiance and the areas of the object faces reveals the compact form of Eq 2.35 for convex objects for any BRDF:

$$\hat{I}_{convex} = G \vec{a}. \quad (2.114)$$

Given a light curve, direct shape inversion schemes sample m candidate normal vectors \hat{n} on the unit sphere to fit an EGI to the observed light curve $\hat{I}_{\text{obs}} \in \mathbb{R}^{n \times 1}$ [5], [12]. 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 (NNLS) problem and can be solved efficiently for large numbers of normal vectors and light curve data points \hat{I}_{obs} :

$$\min_{\vec{a}} \|\hat{I}_{\text{obs}} - G\vec{a}\|_2 \quad \text{subject to } \vec{a}_i \geq 0, \quad (2.115)$$

to yield an optimal set of face areas \vec{a}_i . In Eq 2.115, G is computed from the chosen normal vectors \hat{n}_i , the uniform BRDF, and the Sun and observer vectors in the body frame of the object. NNLS problems are efficiently solved via Lawson and Hanson's original algorithm [78], or the more recent Fast NNLS (FNNLS) algorithm due to Bro and De Jong [79]. It is important to note that the area estimated with Eq 2.115 is the *albedo-area* due to the reflectivity coefficients in Eq 2.35 or 2.43. If the value of C_d and C_s are uniform with a known ratio C_d/C_s , the recovered geometry will be scaled without impacting the face adjacency or relative feature sizes.

The optimization in Eq. 2.115 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 Fibonacci mapping described by Keinert et al. in [80]. Results are visualized for an icosahedron in the body frame in Figure 2.32. Reconstructing the object at this stage is difficult due to the quantity of faces present in the estimated EGI.

2.6.1.3 EGI Optimization Observability

Depending on the object attitude profile and the geometry of the observer and the Sun, some faces of the object may never be illuminated and observed at the same time. In such cases, the gradient of the light curve error with respect to that normal vector of the EGI is

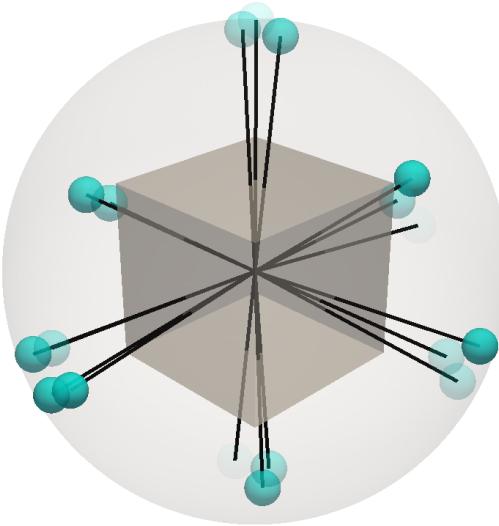


Figure 2.32. Initial EGI optimization example for a cube, using 500 candidate normal vectors, the Phong BRDF with $C_d = 0.5$, $C_s = 0.5$, $n = 10$. Relative face area denoted by higher sphere opacity, reference shape shown in grey.

zero, and that face’s area cannot be recovered. For a given set of candidate normal vectors, the ability to recover a reliable solution from the area optimization problem is known as its observability. As studied by Friedman, the EGI optimization is observable if the Gramian matrix — $G^T G$ in this case — is full rank [12].

This conclusion follows from an unconstrained least squares estimation of the EGI, and is generally equivalent to the statement that the areas of n normal vectors can be estimated if $m \geq n$ measurements are taken with at least one nonzero entry in each column of the reflection matrix G . As noted by Cabrera et al. the nonnegative constraint placed on the face areas a_j mean that the constrained EGI optimization is often estimatable even when the unconstrained form is unobservable [16]. As a result, the unconstrained least squares observability condition is a very conservative lower bound, but is still useful for quantifying the shape information content of the light curve through the rank of $G^T G$. All inversions results presented in this work are performed with observable measurement geometry.

2.6.1.4 EGI Resampling

A normal vector resampling step is presented in this work to promote a more accurate and sparse EGI. The normal vectors used in Fig 2.32 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.

$$\hat{n}_{cone} = \begin{bmatrix} \sqrt{1-z^2} \cos \theta \\ \sqrt{1-z^2} \sin \theta \\ z \end{bmatrix} \quad (2.116)$$

In Eq 2.116 two coordinates are chosen $z \in [\cos \phi, 1]$ and $\theta \in [0, 2\pi)$, yielding a point uniformly distributed on a cone of half-angle ϕ about the central axis $[0, 0, 1]^T$ [81]. These points are then rotated using a direction cosine matrix to center the cone on an axis of interest. The axis of rotation for this transformation is the cross product of the original central axis $[0, 0, 1]^T$ with the final axis \hat{n}_{cone} with the rotation angle θ being the angle between the same two vectors. The principal rotation parameter form of this transformation is given in Eq 2.117 which can be converted into the DCM using Eq 2.7 and 2.5.

$$\begin{aligned} \theta &= \cos^{-1}(\hat{n}_{cone,z}) \\ \lambda &= \hat{n}_{cone} \times [0, 0, 1]^T \end{aligned} \quad (2.117)$$

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. Multiple iterative methods exist for solving nonnegatively

constrained least squares (NNLS) problems. The classical NNLS algorithm was published by Lawson and Hanson and improved later by Bro and De Jong in their Fast NNLS (FNNLS) approach [78], [79].

Existing EGI optimization schemes like those of Fan [5], Friedman [12], and Cabrera [16] are limited by a single normal vector sampling step, leading to a lack of 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. An example of this process is shown in Figure 2.33.

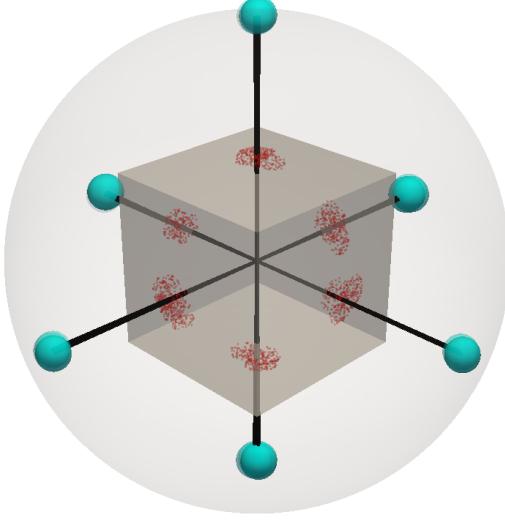


Figure 2.33. Resampled EGI for the cube data using $\phi = \frac{\pi}{20}$, 100 candidate vectors per resampled cone. Relative face area denoted by higher sphere opacity, reference shape shown in grey.

2.6.1.5 EGI Merging

After resampling and reoptimizing with Eq. 2.115, the reestimated EGI is merged by computing all groups \mathcal{G} of EGI vectors within an angular offset α :

$$\mathcal{G}_k = \left\{ \vec{E}_i \in \vec{E} \mid \cos^{-1} \left(\frac{\hat{E}_i \cdot \hat{E}_k}{\|\hat{E}_i\| \|\hat{E}_k\|} \right) < \alpha \right\}. \quad (2.118)$$

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

$$\vec{E}_m = \sum_{\vec{E} \in \mathcal{G}_k} \vec{E} \quad (2.119)$$

Merging the resampled EGI shown in Figure 2.33 produces a final sparse EGI fit for object reconstruction, shown in Figure 2.34.

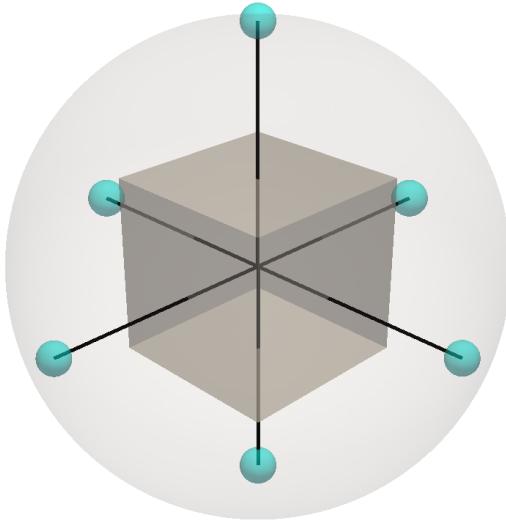


Figure 2.34. Merged EGI for the cube data using $\alpha = \frac{\pi}{10}$. Relative face area denoted by higher sphere opacity, reference shape shown in grey.

2.6.1.6 Geometry Recovery from the EGI

At this stage, the resampled and merged EGI encodes a convex approximation of the underlying object with no guarantee of the closure of this EGI. The EGI closure constraint Eq 2.112 motivates a simple procedure to correct an invalid EGI by adding the mean closure error to each entry:

$$\vec{E}_{\text{closed}} = \vec{E}_{\text{open}} - \sum_{i=1}^m a_i \hat{n}_i. \quad (2.120)$$

The concept of a closure step is not a novel contribution. Fan's method solved an optimization problem to adjust the EGI towards closure [5]. This process is improved 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 through the dual transformation [32]. 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 face of the object:

$$Ax + By + Cz + 1 = 0 \quad (2.121)$$

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 $\vec{h} \in \mathbb{R}^{\|\mathcal{F}\| \times 1}$, where the support vector is the vector of perpendicular distances of each face of the object to the origin. The dual is expressed in terms of the EGI and the supports as:

$$\mathcal{D} = \frac{\vec{E}}{\|\vec{E}\| \vec{h}}. \quad (2.122)$$

The object's vertices \vec{v}_{ref} are found by solving a linear system of equations for each face on the convex hull of dual set vertices. Triplets of vertices on the resulting faces are used to find a single real vertex by intersecting the three planes defining the dual set vertices.

$$\begin{bmatrix} v_{ref,x} \\ v_{ref,y} \\ v_{ref,z} \end{bmatrix} = \begin{bmatrix} v_{i,x} & v_{j,x} & v_{k,x} \\ v_{i,y} & v_{j,y} & v_{k,y} \\ v_{i,z} & v_{j,z} & v_{k,z} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (2.123)$$

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 of the 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.

2.6.1.7 Support Vector Optimization

This work adopts the same objective function proposed by Little for the support vector optimization as used by Fan [5], [32]. Little's objective function leverages work of Minkowski, who proved that any closed EGI is a unique representation of a convex object, up to a translation [32]. In the literature, these convex shapes in \mathbb{R}^3 are known as polytopes. In particular, that unique polytope is merely a scaling and translation from the polytope with volume 1 that minimizes $\vec{h} \cdot \|\vec{E}\|$ [77]. This formulates the support vector optimization problem:

$$\begin{aligned} & \min_{\vec{h}} \left\{ \vec{h} \cdot \|\vec{E}\| \right\} \\ & \text{such that } \frac{1}{3} (\vec{h} \cdot \vec{a}) = 1 \end{aligned} \tag{2.124}$$

Notice that there is a difference between $\|\vec{E}\|$, the areas of the faces on the desired polytope, and \vec{a} , the areas of the polytope recovered by constructing a convex object with the dual transform using Eqs 2.122 and 2.123. This means that a convex object must be fully constructed at each step of the optimization to compute the objective function value. This optimization is highly nonlinear, as adjusting the support of one face may dramatically change the both the areas of surrounding faces as well as the volume of the resulting polytope. Due to the convexity of the problem, an initial guess of $\vec{h} = \mathbf{1}$ is sufficient.

Depending on the optimizer, additional function evaluations will be needed to evaluate support gradient as well as the Jacobian of the constraints. Many optimizers struggle with this optimization problem, either due to an inability to meet the constraint or instability that leads to unbounded supports. Little implemented a constrained gradient descent scheme

from scratch, showing good results for reconstructing simple objects [32]. Gardner solved an equivalent optimization problem using an unspecified solver in MATLAB [82]. In this work, trust-region methods [83] were found to be robust at solving this optimization.

2.6.2 Direct Nonconvex Shape Inversion

2.6.2.1 Nonconvex Feature Detection and Location

Many human-made space objects are, as highlighted in Figure 2.6, highly nonconvex. As a result, their shape inversion is plagued by the fact that the Minkowski problem-driven reconstruction methods of Eq 2.124 cannot recover nonconvex features. Instead of beginning from the ground up, the convex shape guess can be leveraged to detect and locate concavities.

Information about large, unilateral object concavities is obtained during EGI estimation in Eq. 2.115 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. 2.119 is still employed.

The mean axis of prominent concave features is determined by measuring the divergence of the optimized EGI from a closed object with the magnitude of the closure error \vec{e}_{EGI} .

$$\vec{e}_{EGI} = - \sum_{i=1}^m a_i \hat{n}_i. \quad (2.125)$$

The EGI closure error vector in Eq 2.125 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 negligible if there are no concavities present. The closure error may also be negligible if there is no self-shadowing is present over the sampled attitude profile, therefore the closure error merely quantifies the self-shadowing present in the measurements, 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. This is because self-shadowing in that region causes brightness measurements to be smaller than expected for the true area present, leading to an optimized EGI that attributes less area to that section of the object.

As a result, summing the EGI entries produces a vector that points away from the concavity, requiring the negation seen in Eq 2.125 to point towards it.

2.6.2.2 Concavity Creation

Given the direction of a supposed concavity in the body frame of the object, the next challenge is introducing that concavity. This work approximates unknown concave features as valleys with sharp floors, adopting the interior angle ψ between the valley walls as the measure of the feature depth. An interior angle of $\psi = 180^\circ$ implies no concavity, whereas $\psi \rightarrow 0^\circ$ will push the concavity through the object entirely.

The proposed 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 an internal angle. The light curve error of that nonconvex guess is computed, and the internal angle is iterated until a suitable minimum error case is identified.

2.6.2.3 Model Subdivision

Subdividing the initial convex object guess is essential for retaining object detail during concavity creation. A combination of linear subdivision, Loop subdivision, and remeshing algorithms are used to accomplish this. 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 [84]. 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 2.35 to illustrate the available configurations.

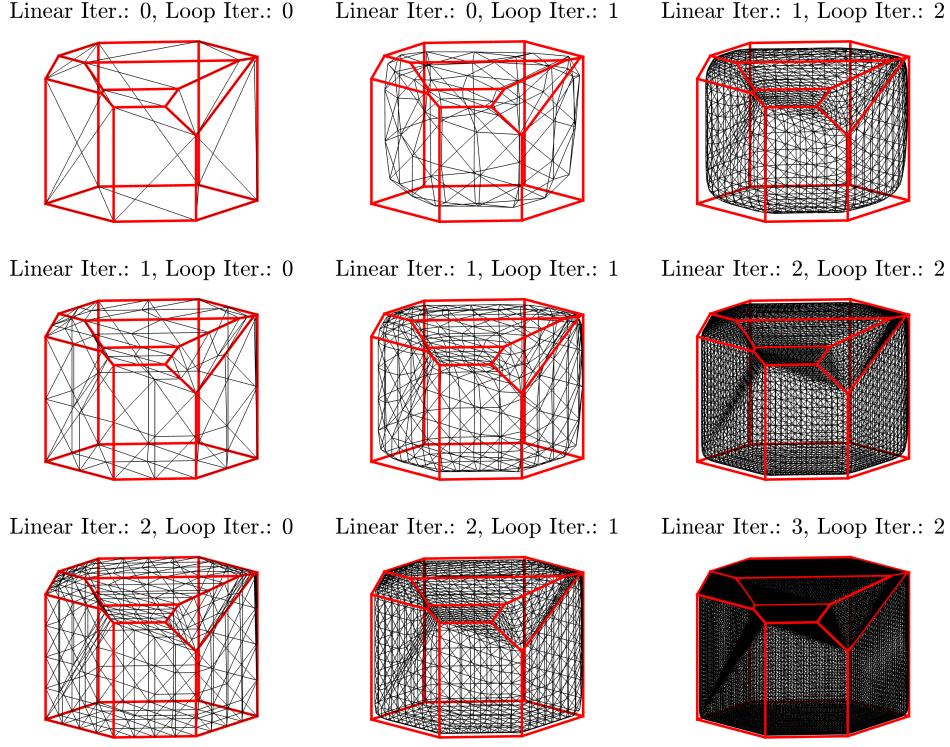


Figure 2.35. Subdivided object (black) with reference (red) with various levels of subdivision

2.6.2.4 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 $\pi/2 - \varepsilon$ to the error vector must be updated, where ε is a bias angle that controls the conservativeness of the classification. Choosing $\varepsilon = 0$ selects all faces above the horizon from the EGI error vector and may lead to concavities that encompass too much of the object. Generally, $\varepsilon \in [0.3, 0.7]$ rad is effective. In reality, all face normals and areas are impacted by the presence of the concavity in the area optimization Eq. 2.115 and EGI correction step Eq 2.120. This bound tends to produce visually accurate concavities. Faces requiring an update are termed *free* faces, with all others termed *root* faces. Explicitly, each face F_i is classified $C(F_i)$ via:

$$C(F_i) = \begin{cases} 0 & \cos^{-1}(\hat{n}_i \cdot \hat{e}_{EGI}) \geq \frac{\pi}{2} - \varepsilon \\ 1 & \cos^{-1}(\hat{n}_i \cdot \hat{e}_{EGI}) < \frac{\pi}{2} - \varepsilon \end{cases}. \quad (2.126)$$

In Eq 2.126, $C(F_i) = 0$ implies a root face, with $C(F_i) = 1$ implying a free face. Vertices on free faces are further classified as being *root-adjacent* or *free*. Root-adjacent vertices are part of at least one root face, whereas free vertices belong to only free faces. The classification of each vertex \vec{v}_i via $C(\vec{v}_i)$ is expressed via:

$$C(\vec{v}_i) = \begin{cases} 0 & \sum_{j \in \mathcal{N}(\vec{v}_i)} C(F_j) = 0 \\ 1 & \sum_{j \in \mathcal{N}(\vec{v}_i)} C(F_j) > 0 \end{cases} \quad \forall i \in \{i \mid C(F_i) = 1\}. \quad (2.127)$$

In Eq 2.127, $\mathcal{N}(\vec{v}_i)$ is the set of face indices including \vec{v}_i as a vertex, yielding $C(\vec{v}_i) = 0$ to imply a free vertex, and $C(\vec{v}_i) = 1$ to imply a root-adjacent face. Classifying vertices in this way results in a border of root-adjacent vertices around the interior free vertices, visualized in Figure 2.36.

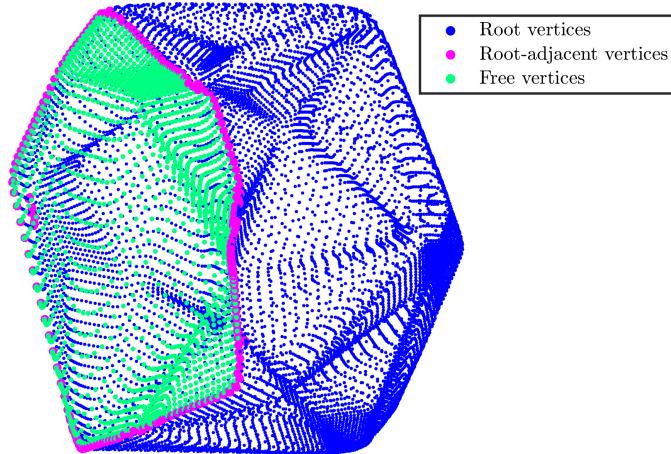


Figure 2.36. Root-adjacent and free vertices

2.6.2.5 Vertex Displacement

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

$$d_i = p_i \sqrt{\csc^2 \frac{\psi_{est}}{2} - 1}, \quad (2.128)$$

where p_i is the distance from each i th free vertex to the nearest root-adjacent vertex.

2.6.2.6 Internal Angle Iteration

Prior work by the author indicated an analytical relationship between this interior angle and the norm of the EGI error vector, summarized in Figure 2.37 [85].

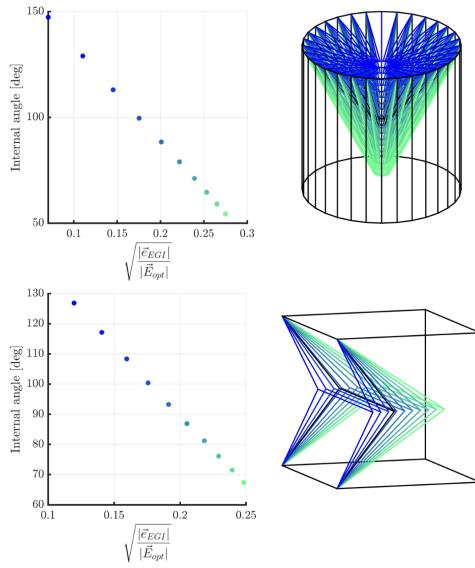


Figure 2.37. EGI error relationship to internal angle for the collapsed cylinder and collapsed house from [85]

The objects in Figure 2.37 were illuminated with a simple Lambertian BRDF and the brightness measurements used to optimize the EGI were produced by randomly sampled Sun and observer vectors. Once specular effects and non-random observation conditions are

accounted for, the linear relationship between ψ and $\sqrt{\frac{\|\vec{e}_{EGI}\|}{\|\vec{E}\|}}$ no longer exists. In short, the optimal concavity angle ψ_{opt} is sought such that:

$$\psi_{opt} = \min_{\psi} (\|\hat{I} - \hat{I}(\psi)\|), \quad (2.129)$$

where \hat{I} is the observed normalized light curve and $\hat{I}(\psi)$ is the light curve resulting with Eq 2.43 using a nonconvex mesh with a concavity of angle ψ introduced. In practice, a line search is sufficient to find the interior angle that minimizes the light curve error of the reconstructed object, summarized in Algorithm 2.

Algorithm 2 Concavity sizing algorithm

```

 $f_{cvx}, v_{cvx}$                                 ▷ Faces and vertices of the convex guess
 $f_{subd}, v_{subd} \leftarrow \text{Subdivide}(f_{cvx}, v_{cvx})$     ▷ Subdivided convex guess
 $\psi = 180^\circ$                                 ▷ Initial guess, no concavity
 $i = 0$                                       ▷ Iteration number
while  $i < i_{max}$  do
     $f_{disp}, v_{disp} = \text{DisplaceVertices}(f_{subd}, v_{subd})$ 
     $\hat{I}_i = \text{LightCurve}(f_{disp}, v_{disp})$ 
     $e_i = \|\hat{I} - \hat{I}_i\|$ 
     $\psi \leftarrow \psi + \Delta\psi$ 
     $i \leftarrow i + 1$ 
end while
 $\psi_{opt} = \min_i e_i$ 

```

2.6.3 Shape Inversion With Noisy Measurements

When the light curve measurements are subject to environmental noise, the shape inversion result — convex or nonconvex — may vary significantly from the noiseless shape estimate. Fan addressed this problem most recently with a multiple hypothesis scheme [5]. Given a number of noisy light curves, Fan produced a convex shape estimate from each and ranked each by its error when simulated against a new set of observations [5]. Fan and Frueh later adapted this scheme with a sequential Monte Carlo method, which began identically by producing a collection of convex guesses from samples of the same noisy light curve **fan20201**. The error in each guess was measured against a new noisy light curve,

enabling a weighted merge of the estimated EGIs, which was then perturbed to produce a new set of estimated shapes [39]. Throughout these methods, Fan assumed that the noise on the light curve was Gaussian and of constant magnitude, although it was noted that the real character of the noise is dependent on both the geometry of the observation and the magnitude of the signal itself [5].

This work presents a method for multiple hypothesis shape inversion that builds on Fan’s work by relaxing the constant, Gaussian noise assumption and is agnostic to the convexity of the candidate objects. This new method also uses an face-dependent estimate of the uncertainty in each candidate shape’s geometry, yielding final object estimates that take only the most accurate features from each candidate object.

2.6.3.1 Weighted Shape Interpolation

To develop this multiple hypothesis inversion procedure that works for both convex and nonconvex shape candidates, it is useful to be able to take a weighted average of an arbitrary set of triangle meshes. Given candidate object meshes M_i with scalars w_i , this weighted average process should be of the form:

$$M_{\text{interp}} = \sum_i w_i M_i. \quad (2.130)$$

The algorithm presented in this work for weighted mesh averaging depends on signed distance fields (SDFs). The SDF implicitly represents a shape by associating each point $\mathbf{x} \in \mathbb{R}^3$ with the distance to the closest point on the surface of the object [86]. As a result, the gradient of the SDF points directly away from the closest piece of the object’s geometry. For a given SDF $f(\mathbf{x})$, the surface of the object S_{obj} is given by all points \mathbf{x} that yield $f(\mathbf{x}) = 0$:

$$S_{\text{obj}} = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) = 0 \right\}. \quad (2.131)$$

Computing the SDF of a triangulated mesh breaks down into computing distances from the points, line segments, and planes making up the mesh to the queried point [86]. A slice of the SDF of a test model is displayed in Figure 2.38.

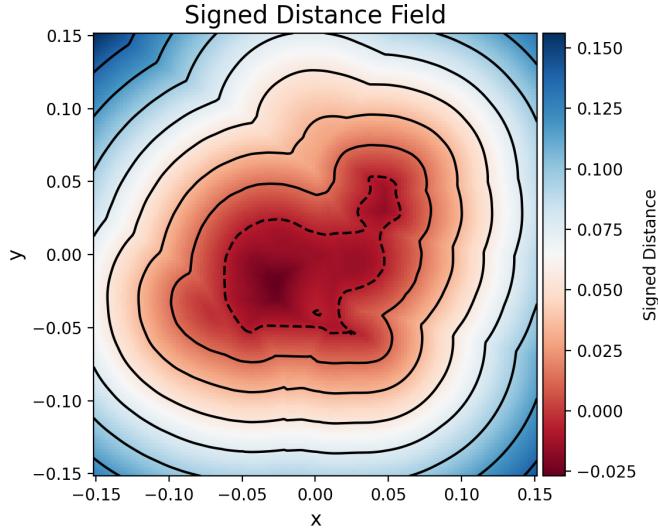


Figure 2.38. SDF slice of the Stanford bunny model

Interpolating three-dimensional meshes using signed distance fields is not a novel concept of this work. Cohen-Or et al. used distance fields with anchor points to find warping functions between two geometries [87]. A simple interpolation strategy between two shapes can be accomplished through a weighted average of the respective SDFs. If both objects are weighted at 50%, the weighted sum of their SDFs produces a surface that lies halfway in between the two original shapes, measured perpendicular to the input objects' surfaces. This interpolated zero level set surface can be extracted through any three dimensional isocontouring algorithm such as marching cubes [88] or flying edges [89]. For example, interpolating between a torus and an icosahedron using this method yields Figure 2.39.

The proposed algorithm for shape interpolation for SDF interpolation of two shapes M_1, M_2 with convex weights w_1, w_2 is:

Algorithm 3 can operate on arbitrary convex or nonconvex meshes, making it well suited to interpolating between light curve inversion results.

Algorithm 3 SDF interpolation

Require: $w_1 + w_2 = 1$

$\text{SDF}_{\text{interp}} \leftarrow w_1 \cdot \text{SDF}_{M_1} + w_2 \cdot \text{SDF}_{M_2}$
 $M_{\text{interp}} = \text{Isocontour}(\text{SDF}_{\text{interp}}, 0)$.

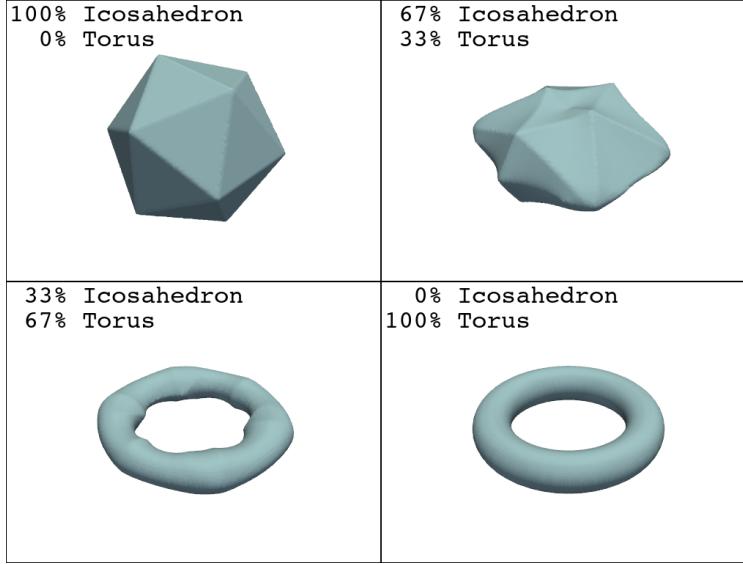


Figure 2.39. SDF interpolation between a torus and an icosahedron

The procedure in Algorithm 3 can be further generalized by allowing the weights w_i to be functions of the location in \mathbb{R}^3 of each SDF grid point:

$$\text{SDF}_{\text{interp}}(x, y, z) = w_1(x, y, z)\text{SDF}_{M_1}(x, y, z) + w_2 \cdot \text{SDF}_{M_2}(x, y, z). \quad (2.132)$$

To discourage degenerate behavior, it is necessary to enforce $w_i(x, y, z) > 0 \forall (x, y, z) \in \mathbb{R}^3$.

2.6.3.2 Shape Estimate Uncertainty

When merging multiple inaccurate shape guesses it is advantageous to measure the uncertainty in each candidate shape and use that uncertainty as a weighting to bias the merging process. The formulation used in this work relies on two face-wise uncertainty terms. First, a face should have high uncertainty if very little light would have reflected off that face to contribute to the light curve. Second, that face should also have high uncertainty if its reflected light has highly contaminated by background noise. As a result, the only faces of an estimated shape with low uncertainty should be those that reflect a large *quantity* of light, increasing the observability of that face, while being uncontaminated by the background

to yield a high *quality* signal. These uncertainty terms are computed by first scaling the columns of the reflection matrix defined in Eq 2.113 by the area of each respective face a_j :

$$H_{ij} = a_j G_{ij}. \quad (2.133)$$

The resulting matrix H has the same dimensions as G , but has entries H_{ij} representing the normalized irradiance expected from face j and time i . The total normalized irradiance over all timesteps for each face is given by:

$$\hat{I}_j = \sum_i H_{ij}. \quad (2.134)$$

Computing \hat{I}_j is useful as it represents the total quantity of signal expected from face j across the light curve. Using this value, the data quantity weighting is:

$$u_{quant,j} = 1 - \frac{\hat{I}_j - \min_j \hat{I}_j}{\max_j \hat{I}_j - \min_j \hat{I}_j}. \quad (2.135)$$

Via Eq 2.135, the face with the most influence on the overall light curve is assigned a data quantity uncertainty of $u_{quant,j} = 0$ while the face with the least influence is assigned $u_{quant,j} = 1$. As discussed, a high quantity of available data does not necessarily mean that the data has a high SNR. Estimating the SNR in the measured counts is certainly possible, especially if the observing station is well characterized, but is a strong assumption for an unknown ground station. It would be preferable to estimate the variance of the measured signal through the convex shape guess itself. As the NNLS optimization of the EGI is guaranteed to produce the convex shape that best matches the measured irradiances, the convex guess is a good estimator for the mean of the received signal so long as the noise on the measurements is zero mean. As a result, the variance of the light curve error:

$$\text{Var}(\delta \hat{I}) = \text{Var}(Ga - \hat{I}_{meas}) \quad (2.136)$$

generally matches the true signal and noise variance. This means that scaling the expected normalized irradiance H_{ij} by the absolute value of the light curve error yields a

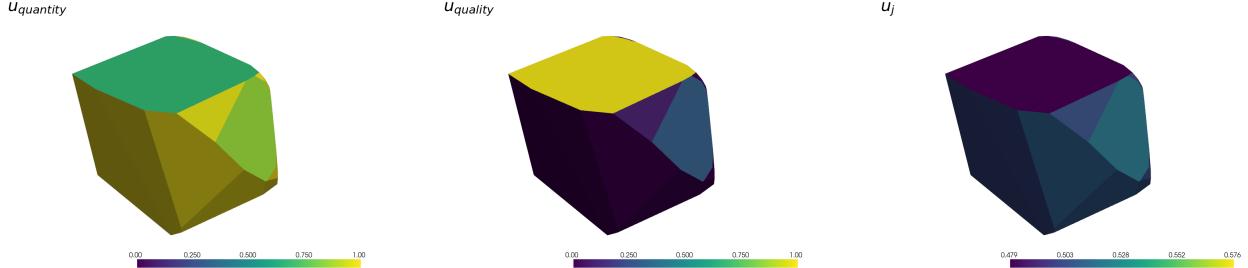


Figure 2.40. Quantity (left), quality (middle), and overall face uncertainty (right) weights

quantity that is proportional to the number of false counts above or below the true mean attributed to a certain face:

$$\delta \hat{I}_{tot,j} = a_j \sum_i (G_{ij} a_j - \hat{I}_{meas,i}) H_{ij}, \quad (2.137)$$

yielding a weighting for the data quality attributed to each face of the convex guess:

$$u_{qual,j} = \frac{\delta \hat{I}_{tot,j} - \min_j \delta \hat{I}_{tot,j}}{\max_j \delta \hat{I}_{tot,j} - \min_j \delta \hat{I}_{tot,j}}. \quad (2.138)$$

These two uncertainty weightings are combined at each face for a global uncertainty estimate:

$$u_j = \frac{u_{qual,j}}{2} + \frac{u_{quant,j}}{2}. \quad (2.139)$$

Eq 2.139 is formulated such that a face is only assigned a low uncertainty if it has both high data quality and quantity. This method conservatively discounts low quantities of high quality data as well high quantities of low quality data. Figure 2.40 displays the quality, quantity, and overall uncertainty weighting for the faces of a convex shape guess.

Figure 2.40 conveys that u_{qual} and u_{quant} convey independent information about each face depending on the measured light curve and its noise — higher data quantity does not necessarily imply higher quality.

2.6.3.3 Spherical Nearest Neighbor Weights

One final challenge remains: translating the uncertainties of the faces of an object to a weighting function that can be applied throughout \mathbb{R}^3 to weight the SDF. Simply projecting points onto the unit sphere and performing nearest-neighbor interpolation accomplishes this task well. For a set of reference points \mathbf{r}_i each with a weight w_i , the weight w_q of query point $\mathbf{r}_q \in \mathbb{R}^3$ is the solution to:

$$w_q = \min_i (\|\mathbf{r}_q - \mathbf{r}_i\|). \quad (2.140)$$

In practice, this optimization problem is solved with a Balltree, a spatial search data structure designed to efficiently find nearest neighbors in n -dimensional space bounded by the $n - 1$ -dimensional sphere [90]. The results of this nearest neighbor interpolation for a number of query points is shown in Figure 2.41.

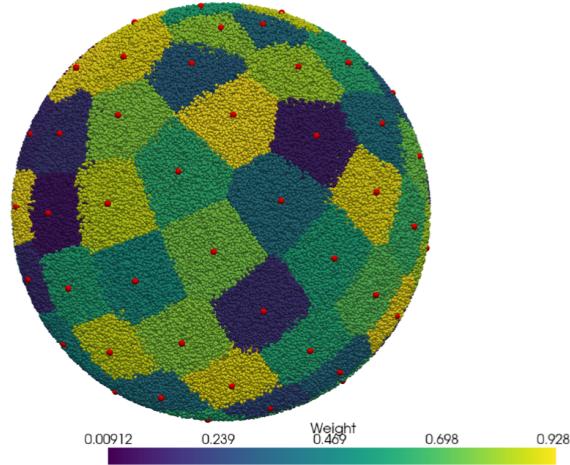


Figure 2.41. Nearest neighbor interpolation between reference points (red) with uniformly distributed weights on $[0, 1]$.

2.6.3.4 Full Noisy Inversion Procedure

Given n noisy light curves from an object of unknown shape, the procedure determining a single best estimate of the object's true shape has three stages.

1. Convex shape guesses are computed for each noisy light curve.
2. Each shape is checked for potential nonconvex features by introducing singular concavities in the direction of the EGI error vector following Algorithm 2. Each nonconvex guess along with the convex guess is simulated in the same observation conditions to compute an error with respect to the original light curve. The guess — convex or nonconvex — with the lowest error is selected moving forward.
3. Now with n convex or nonconvex shape estimates, one for each original light curve, the uncertainty in each face of each object is computed with Eq 2.139.
4. Given these uncertainties, each object is assigned a spherical weighting function $w_i(x, y, z)$.
5. The objects are merged together via their weights using the SDF procedure outlined by Algorithm 3, yielding a final shape estimate.

3. Simulated Light Curve Results

3.1 Accurate Satellite Models

To produce accurate light curves for real space objects, 3D models files were collected and tuned for use with the light curve simulator. These models are shown in Figure 3.1. Each model is correctly scaled, with accurate material properties applied to their faces. Figure 3.1 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.

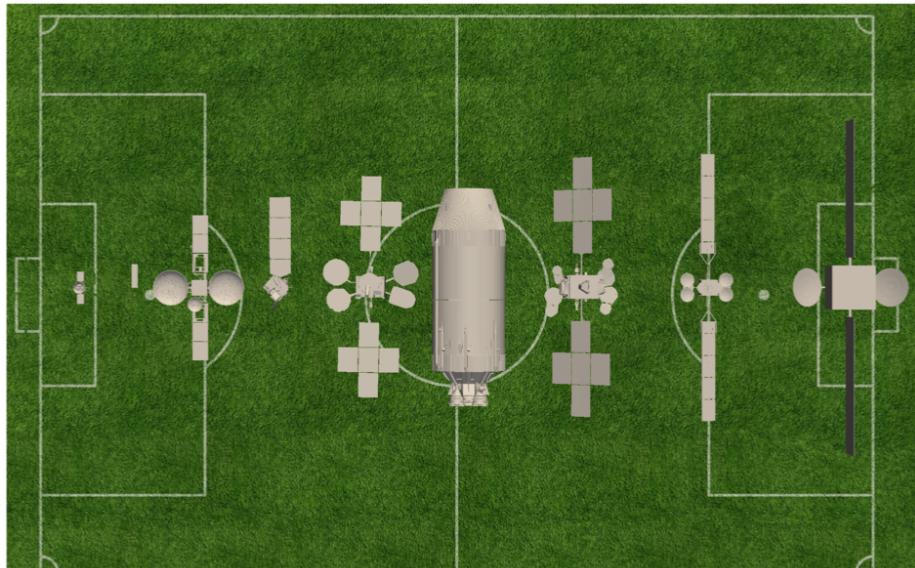


Figure 3.1. Selected space objects with soccer field for size reference. In order, the objects are TESS, Starlink V1, TDRS, Landsat 8, Hispasat 30W-6, Saturn V SII, TELSTAR 19V, HYLAS 4, and simplified ASTRA.

3.2 Noiseless Light Curves

TODO: this

3.3 Realistic Light Curves

TODO: this

4. Shape Inversion Results

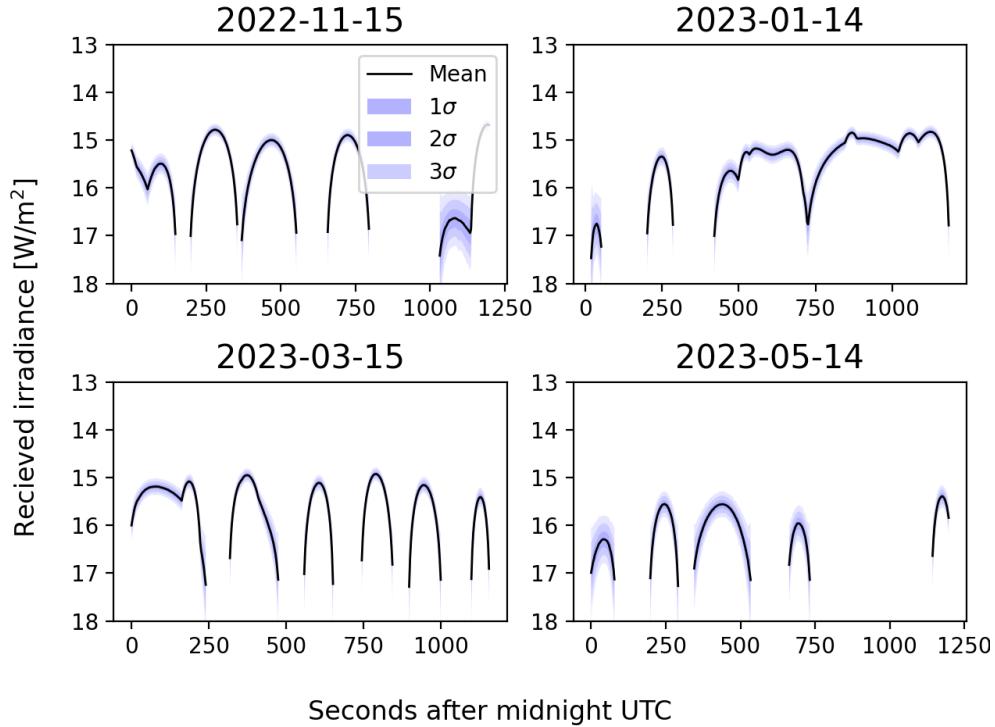


Figure 4.1. Light curves for a 1-meter diffuse cube with $C_d = 0.5$ observed from the Purdue Optical Ground Station at $32.900^\circ \text{ N}, -105.533^\circ \text{ W}$. Object is in the orbit of GOES 15 under torque-free rigid body motion with an inertia tensor $J = \text{diag}(1.0, 2.0, 3.0) [\text{kg} \cdot \text{m}^2]$, $q_0 = [0, 0, 0, 1]^T$, $\omega_0 = [0.01, 0.02, 0.01] [\text{rad}/\text{s}]$

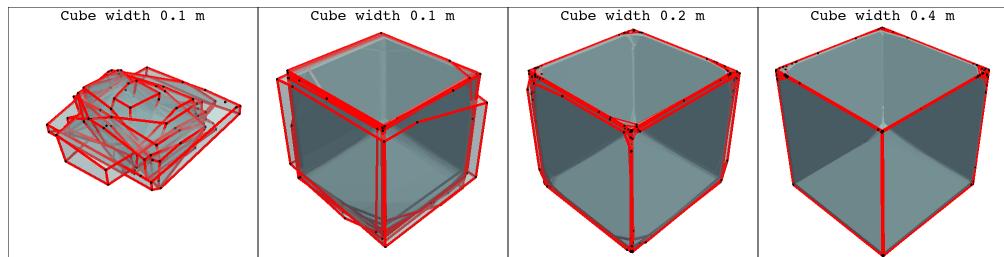


Figure 4.2. Convex shape inversions of a cube from noisy measurements

4.1 Convex Shape Inversion Without Noise

4.1.1 Object Reconstruction Improvements

The proposed resampling and merging steps in the object reconstruction process produce sparser results, in less computation time, with fewer convergence issues. Figure 4.3 shows object reconstructions with each available EGI. The resampled EGIs does not converge in the maximum number of iterations evaluations (100) due to the density of faces with nearby normal vectors, causing large linearization errors the gradient estimation that lead to inefficient optimizer steps. It is clear that the final merged EGI produces the most accurate reconstruction of the truth object.

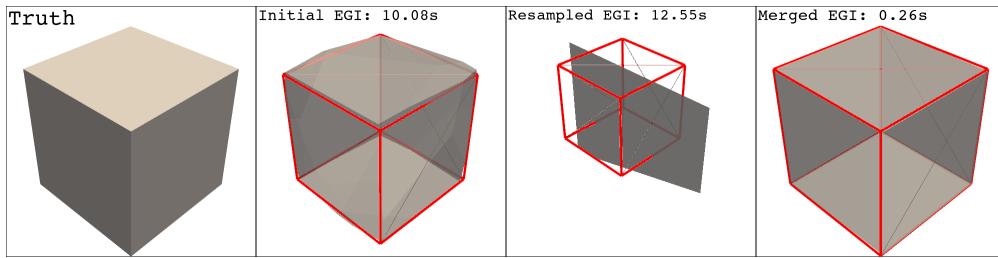


Figure 4.3. Shape inversions from the initial, resampled, and merged EGIs

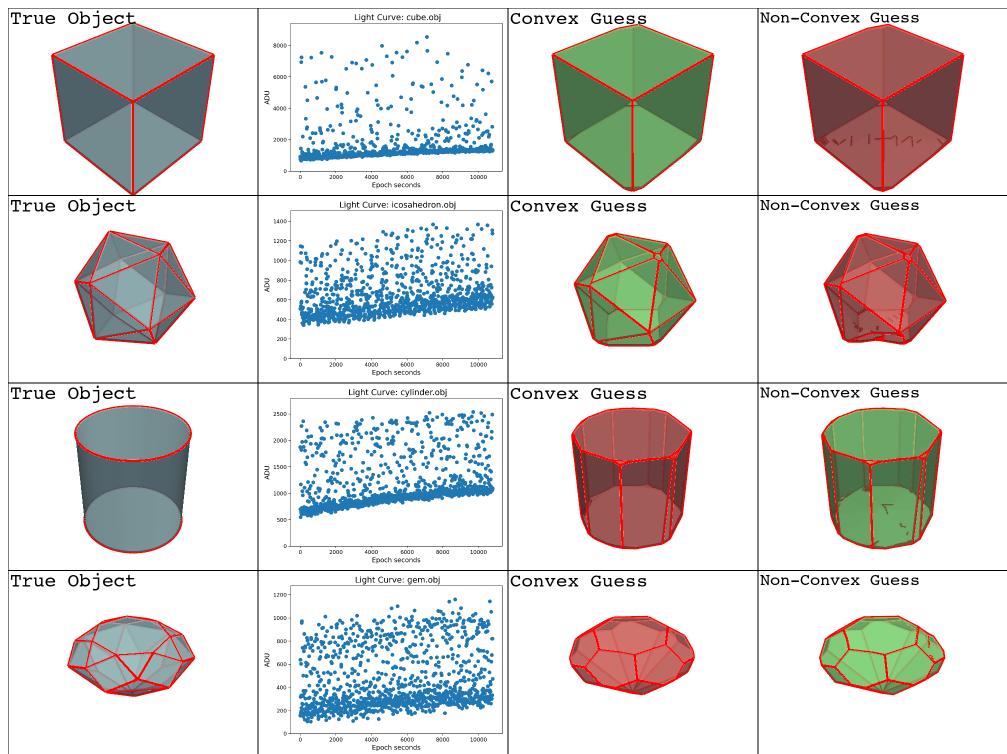


Figure 4.4. Shape inversion results for convex objects without noise in the light curve

4.2 Nonconvex Shape Inversion Without Noise

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 4.4.

The collapsed cube and icosahedron in Figure 4.4 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 $\psi = 180^\circ$. This is illustrated in Figure ?? using the same input convex objects and attitude profiles as in Figure ??.

Figure ?? clearly displays the compatibility of vertex displacement with truly convex objects. All objects are reconstructed faithfully in both their convex and nonconvex 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.

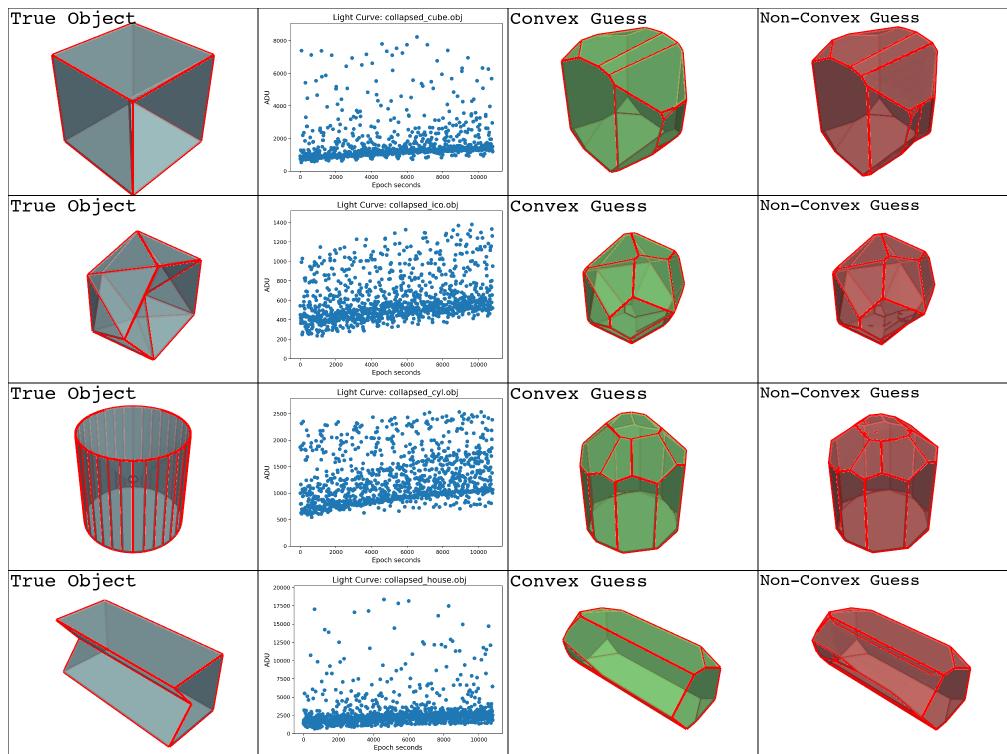


Figure 4.5. Shape inversion results for nonconvex objects without noise in the light curve

4.3 Convex Shape Inversion With Noise

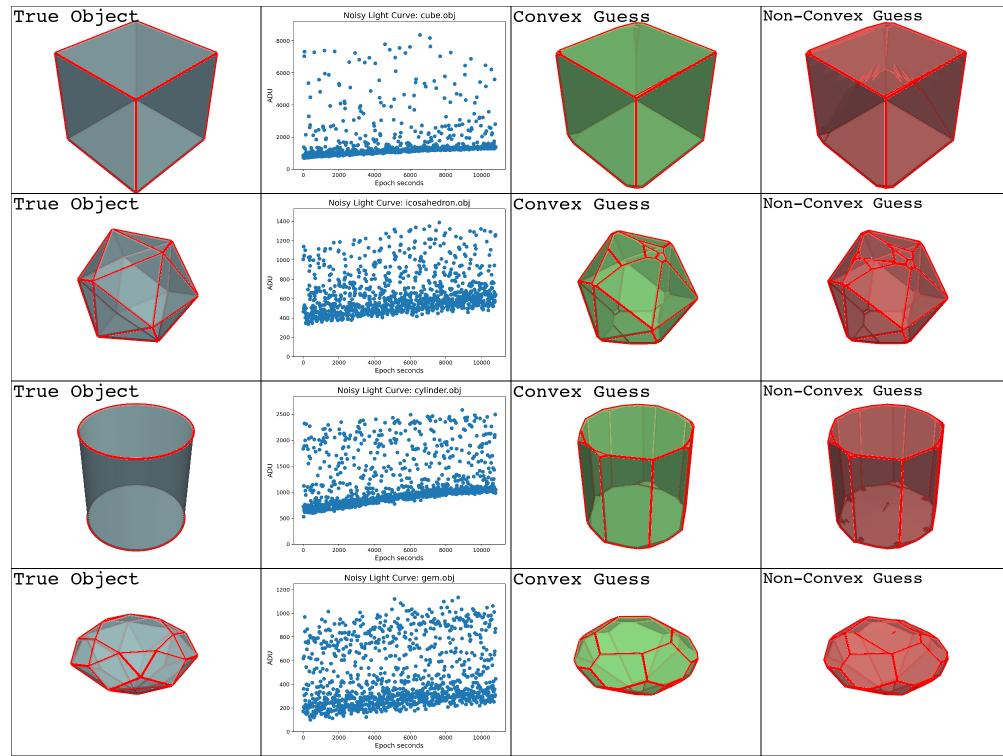


Figure 4.6. Shape inversion results for convex objects with noise in the light curve

TODO: this

4.4 Nonconvex Shape Inversion With Noise

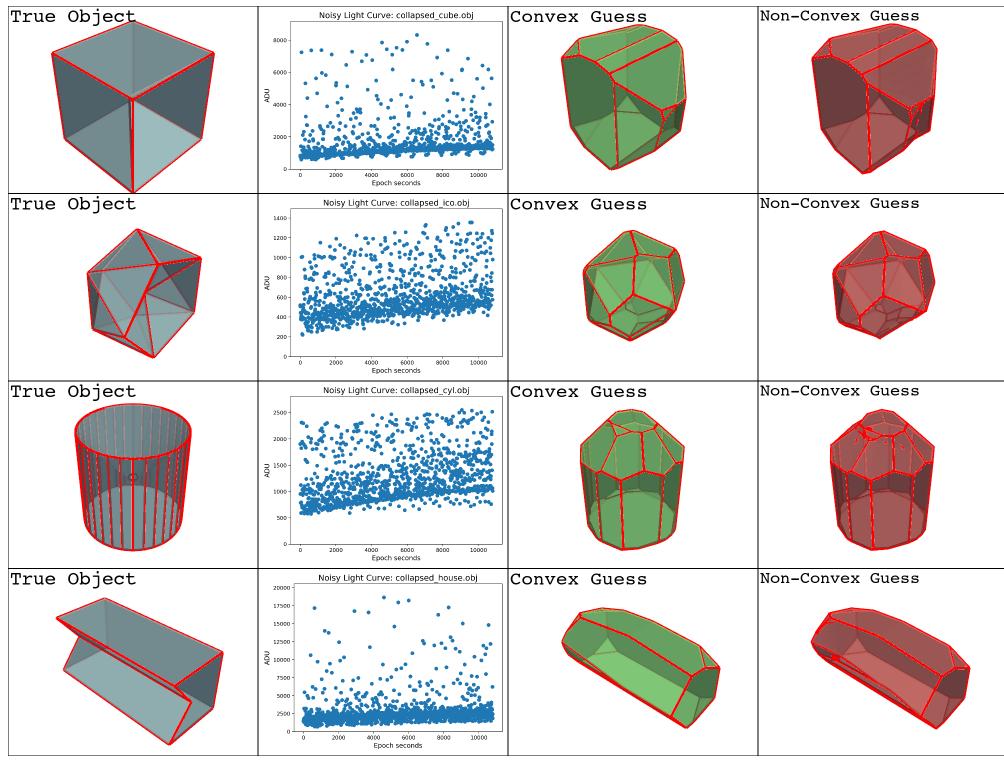


Figure 4.7. Shape inversion results for nonconvex objects with noise in the light curve

TODO: this

4.5 Multiple Hypothesis Inversion

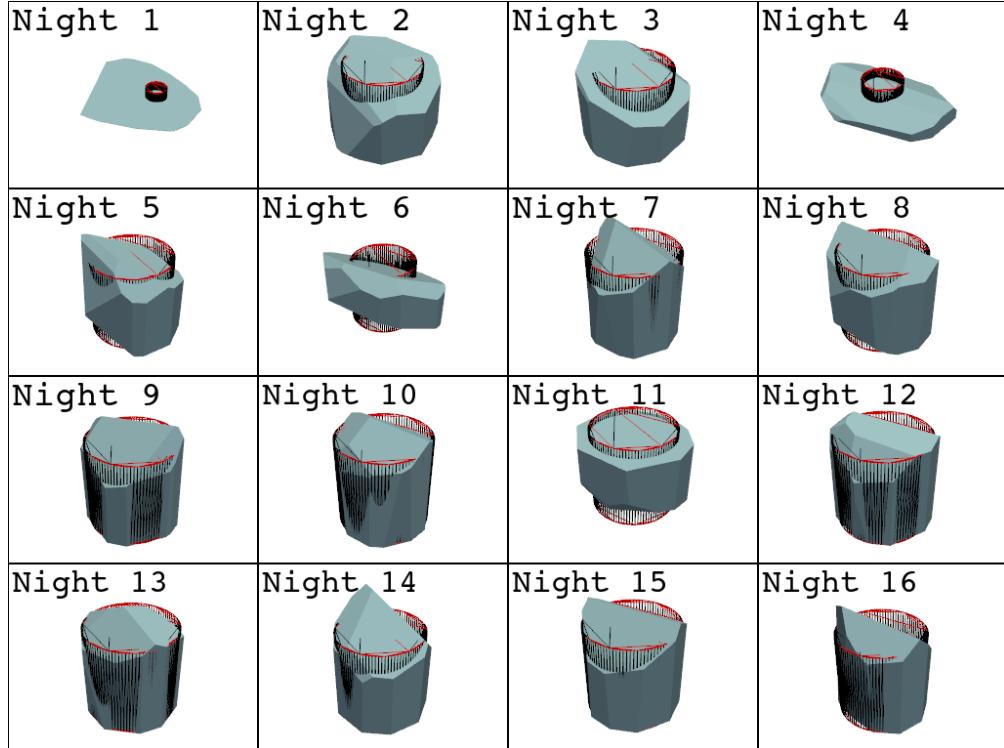


Figure 4.8. Multiple hypothesis candidates from 16 nights of noisy observations of a 1-meter cylinder in the orbit of GOES 15. The object is simulated with inverted with a Phong BRDF using $C_d = 0.5$, $C_s = 0.5$, $n = 10$. Nightly light curves are each 3 hours long with data collected every 10 seconds, beginning January 1, 2023, 05:00 UTC.

5. Recommendations

6. Future Work

The next step with this work would be to extensively validate the CCD performance model against real observations taken by the Purdue Optical Ground Station. This would involve quantifying the sensor noise sources through zero second exposures and closed shutter exposures to isolate the effects of the dark noise and readout noise. Next, the background model would be fit to an array of observations at different times of the night, zenith angles, and lunar phases. Once both the background and sensor effects are well calibrated, the overall CCD model can be tested against the observed signal counts and light curve SNR of known objects.

More work must also be done to perform light curve inversion on actively-controlled box-wing satellites. This would involve a simple adaptation of the EGI inversion method where some normal vectors are allowed to move in the body frame, enabling solar panels to track the Sun. Such a parametric approach could be used to check whether the light curve of an unknown object — in an unknown attitude profile — fits the expected light curve of a box-wing satellite with Sun tracking solar panels.

Finally, investigating shape inversion with multispectral measurements may open new avenues for simultaneously estimating attitude or material properties in addition to a convex shape. This might be accomplished operationally by switching filters between images, or by using a dedicated multispectral sensor. In either case, adding a second axis to the available data would enable far more robust characterization of space objects with fewer assumptions.

7. Appendices

7.1 Shadow Mapping

7.1.1 Shading

Algorithm 4 Pixel-wise shading algorithm with shadow mapping

$L \in \mathbb{S}^2$	▷ Unit vector from object origin towards Sun
$O \in \mathbb{S}^2$	▷ Unit vector from object origin towards observer
$N \in \mathbb{S}^2$	▷ Outward-pointing surface normal vector at pixel coordinates
$(C_d, C_s, n) \in \mathbb{S}^2$	▷ Reflection coefficients and exponent for the BRDF
Require: $C_d + C_s \leq 1$	▷ Enforce energy conservation
$MVP_{Sun} \in \mathbb{R}^{4 \times 4}$	▷ MVP matrix for the Sun camera
$(x, y) \in \mathbb{Z}^2$	▷ Integer pixel coordinates from the observer camera
$R_{pix,obs} \in \mathbb{R}^3$	▷ World coordinates of the pixel; provided by OpenGL
$[(x_{homo,Sun}, y_{homo,Sun}, \dots)] \leftarrow MVP_{Sun} [R_{pix,obs}, 1]^T$	
$x_{Sun} \leftarrow (1 + p_{x,homo}) \frac{w_{pix}}{2}$	▷ Homogeneous coordinates from the Sun camera
$y_{Sun} \leftarrow (1 + p_{y,homo}) \frac{a \cdot \bar{w}_{pix}}{2}$	
$D_{Sun} \leftarrow d(x_{Sun}, y_{Sun})$	▷ Closest pixel depth to the Sun direction
$D_{obs} \leftarrow (L - O) \cdot L$	▷ Closest pixel depth in the Observer direction
if $D_{obs} > D_{Sun}$ then	
$\delta_{ss} = 1$	▷ Pixel is self-shadowed
else	
$\delta_{ss} = 0$	▷ Pixel may be illuminated
end if	
if $(N \cdot L) > 0$ and $(N \cdot O) > 0$ then	
$f_r(\mathbf{x}, L, O) = 0$	▷ Pixel cannot be both observed and illuminated
else	
$f_r(\mathbf{x}, L, O) = \text{Phong}(L, O, N, C_d, C_s, n)$	▷ The pixel is shaded with the BRDF
end if	
$\text{IM}(x, y) = f_r(\mathbf{x}, L, O) (N \cdot L)$	▷ Image pixel value

7.2 Astronomical Spectra Data

7.2.0.1 Atmospheric Extinction

Data from [9]. The atmospheric extinction coefficient is dimensionless.

```
{"lambda": [0.0, 3.2e-07, 3.400000000000003e-07, 3.6e-07, 3.79999999999996e-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]}

---


```

7.2.1 Lunar Phase Factor

The lunar phase factor is a function of the phase angle in radians and is dimensionless.

Data from [67].

```
{"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]}

---


```

7.2.2 Scattered Moonlight

The scattered moonlight radiance in $\left[\frac{W}{sr \cdot m^2 \cdot m} \right]$ is a function of the difference in the line of sight and Moon azimuths `delta_az` in radians, the zenith angle of the moon `z_moon` in radians, and the zenith angle of the line of sight `z_obs` in radians. Data from [67].

```
{"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, 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]], [[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]]]}
```

7.2.3 Zodiacal Light

The zodiacal light surface brightness in S_{10} is a function of the latitude "ecliptic_lat" and longitude "ecliptic_lon" of the line of sight in the solar system ecliptic reference frame, both expressed in radians. Data from [74].

```
{"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, 247.0000000000003, 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, 220.0000000000003, 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, 224.0000000000003, 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, 146.0, 150.0], [110.0000000000001, 112.0000000000001, 112.0000000000001, 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], [103.0, 105.0, 105.0, 108.0, 111.0000000000001, 111.0000000000001, 106.0, 107.0, 107.0, 108.0, 107.0, 111.0000000000001, 107.0, 106.0, 108.0, 108.0, 108.0, 108.0, 108.0]]]}
```

7.3 Telescope Parameters

7.3.0.1 Purdue Optical Ground Station

Parameter	Value
FWHM	1.5
Sensor dimensions	$0.03690 \times 0.03690 [m]$
<i>f</i> number	7.2
Aperture diameter D	0.35560 [m]
Secondary diameter	0.1724660 [m]
Sensor pixels	4096×4096
Pixel size	$9.009 \cdot 10^{-6} [m/\text{pix}]$
Pixel scale s_{pix}	0.72545 [arcsec]
Field of view	$0.824425^\circ \times 0.824425^\circ$
Integration time Δt	10 [s]
Integration dark noise λ_{dark}	3 [$\text{e}^-/\text{pix}/\text{s}$]
Read noise variance σ_{read}^2	9 [e^-]
CCD gain g	1

Table 7.1. Purdue Optical Ground Station telescope parameters

7.4 Wavefront OBJ Example

```
# Blender v2.92.0 OBJ File: ''
# www.blender.org
mtllib cube.mtl
o Cube_Cube.003
v 1.000000 1.000000 -1.000000
v 1.000000 1.000000 1.000000
v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 1.000000 -1.000000
v -1.000000 1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v -1.000000 -1.000000 1.000000
vt 0.625000 0.000000
vt 0.375000 0.250000
vt 0.375000 0.000000
vt 0.625000 0.250000
vt 0.375000 0.500000
vt 0.625000 0.500000
vt 0.375000 0.750000
vt 0.625000 0.750000
vt 0.375000 1.000000
vt 0.125000 0.750000
vt 0.125000 0.500000
vt 0.875000 0.500000
vt 0.625000 1.000000
vt 0.875000 0.750000
vn 1.0000 0.0000 0.0000
vn 0.0000 -1.0000 0.0000
vn -1.0000 0.0000 0.0000
vn 0.0000 1.0000 0.0000
vn 0.0000 0.0000 -1.0000
vn 0.0000 0.0000 1.0000
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] ESA Space Debris Office, “Esa’s annual space environment report,” European Space Agency, Tech. Rep., Apr. 2022.
- [2] N. G. S. F. Center, *Vanguard 1*, <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1958-002B>, Accessed: 2023-09-04.
- [3] D. Vallado, *Fundamentals of Astrodynamics and Applications*, 4th ed. J. Wertz, Ed. Microcosm Press, Mar. 2013.
- [4] C. Frueh, *Space traffic management*, Lecture Notes, 2019.
- [5] 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](https://doi.org/10.25394/PGS.12749570.v1).
- [6] T. Schildknecht, R. Musci, and M. Fruh C. Ploner, “Color Photometry and Light Curve Observations of Space Debris in GEO,” in *Advanced Maui Optical and Space Surveillance Technologies Conference*, C. Paxson, H. Snell, J. Griffin, et al., Eds., Jan. 2008, E51, E51.
- [7] T. Schildknecht, N. Koshkin, E. Korobeinikova, et al., “Photometric Monitoring of Non-resolved Space Debris and Databases of Optical Light Curves,” in *Advanced Maui Optical and Space Surveillance Technologies Conference*, Jan. 2015, 25, p. 25.
- [8] A. Burton and S. Fan, “Mapping the solution space for light curve attitude estimation,” *43rd COSPAR Scientific Assembly. Held 28 January-4 February*, vol. 43, p. 2193, 2021.
- [9] 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.
- [10] 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>.

- [11] 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](https://doi.org/10.2514/6.2016-5504).
- [12] A. M. Friedman, “Observability analysis for space situational awareness,” Ph.D. dissertation, Purdue University, Apr. 2020. doi: [10.25394/PGS.12196863.v1](https://doi.org/10.25394/PGS.12196863.v1).
- [13] 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.
- [14] 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>.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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/>.

- [21] 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](https://doi.org/10.48550/ARXIV.2004.12270). [Online]. Available: <https://arxiv.org/abs/2004.12270>.
- [22] 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](https://doi.org/10.1016/j.actaastro.2021.01.048).
- [23] T. Möller and B. Trumbore, “Fast, minimum storage ray-triangle intersection,” *Journal of Graphics Tools*, vol. 2, Aug. 2005. DOI: [10.1145/1198555.1198746](https://doi.org/10.1145/1198555.1198746).
- [24] 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](https://doi.org/10.1145/280811.280980).
- [25] M. Ashikhmin and P. Shirley, “An anisotropic phong brdf model,” *Journal of Graphics Tools*, vol. 5, Jan. 2001. DOI: [10.1080/10867651.2000.1048752](https://doi.org/10.1080/10867651.2000.1048752).
- [26] R. L. Cook and K. E. Torrance, “A reflectance model for computer graphics,” *ACM Trans. Graph.*, vol. 1, no. 1, pp. 7–24, Jan. 1982, ISSN: 0730-0301. DOI: [10.1145/357290.357293](https://doi.org/10.1145/357290.357293). [Online]. Available: <https://doi.org/10.1145/357290.357293>.
- [27] M. Oren and S. K. Nayar, “Generalization of lambert’s reflectance model,” in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’94, New York, NY, USA: Association for Computing Machinery, 1994, pp. 239–246, ISBN: 0897916670. DOI: [10.1145/192161.192213](https://doi.org/10.1145/192161.192213). [Online]. Available: <https://doi.org/10.1145/192161.192213>.
- [28] W. Matusik, H. Pfister, M. Brand, and L. McMillan, “A data-driven reflectance model,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 759–769, Jul. 2003.
- [29] H. N. Russell, “On the light-variations of asteroids and satellites,” *The Astrophysical Journal*, vol. 24, no. 1, pp. 1–18, 1906.
- [30] M. Kaasalainen and J. Torppa, “Optimization methods for asteroid lightcurve inversion: I. shape determination,” *Icarus*, vol. 153, no. 1, pp. 24–36, 2001, ISSN: 0019-1035. DOI: <https://doi.org/10.1006/icar.2001.6673>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0019103501966734>.

- [31] 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.
- [32] 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.
- [33] J. Durech 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](https://doi.org/10.1051/0004-6361:20030505).
- [34] 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](https://doi.org/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>.
- [35] J. Durech, V. Sidorin, and M. Kaasalainen, “DAMIT: a database of asteroid models,” *Astronomy and Astrophysics*, vol. 513, A46, A46, Apr. 2010. DOI: [10.1051/0004-6361/200912693](https://doi.org/10.1051/0004-6361/200912693).
- [36] M. Kaasalainen, J. urech, and V. Sidorin, *DAMIT: Database of Asteroid Models from Inversion Techniques*, Astrophysics Source Code Library, record ascl:1412.004, Dec. 2014. ascl: [1412.004](https://doi.org/10.17822/1412.004).
- [37] 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](https://doi.org/10.1117/12.683015). [Online]. Available: <https://doi.org/10.1117/12.683015>.
- [38] 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](https://doi.org/10.1007/s40295-019-00190-3).
- [39] 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.

- [40] 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](https://doi.org/10.1007/s40295-021-00293-w).
- [41] 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.
- [42] 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](https://doi.org/10.2514/1.62986).
- [43] 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.
- [44] 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.
- [45] S. Karpov, E. Katkova, G. Beskin, *et al.*, “Massive photometry of low-altitude artificial satellites on Mini-Mega-TORTORA,” in *Revista Mexicana de Astronomia y Astrofisica Conference Series*, ser. Revista Mexicana de Astronomia y Astrofisica Conference Series, vol. 48, Dec. 2016, pp. 112–113.
- [46] C. Benson, D. Scheeres, and N. Moskovitz, “Light curves of retired geosynchronous satellites,” in *7th European Conference on Space Debris*, vol. 7, 2017, p. 04.
- [47] N. Koshkin, L. Shakun, E. Korobeynikova, *et al.*, “Monitoring of space debris rotation based on photometry,” *Odessa Astronomical Publications*, vol. 31, pp. 179–185, 2018.
- [48] Y. Wang, X. Du, J. Zhao, Z. Yin, and Y. Song, “What causes the abrupt changes in the light curve of a geo satellite?” *Acta Astronautica*, vol. 153, pp. 130–137, 2018, ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2018.10.035>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576518313857>.
- [49] Space-Track.org, *Space-track.org*, <https://www.space-track.org/>, Accessed: 2023-10-10, 2023.

- [50] T. Payne, F. Hoots, A. Butkus, Z. Slatton, and D. Nguyen, “Improvements to the sgp4 propagator (sgp4-xp),” in *Proceedings of the 2022 AMOS Technical Conference*, 2022.
- [51] C. J. Benson, D. J. Scheeres, W. H. Ryan, and E. V. Ryan, “Cyclic complex spin state evolution of defunct geo satellites,” in *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI*, 2018.
- [52] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. Springer, May 2014.
- [53] M. D. Shuster, “Survey of attitude representations,” *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, Oct. 1993.
- [54] *Wavefront obj file format*, <https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml>, Accessed: 2023-08-30, 2020.
- [55] 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](https://doi.org/10.1145/2513456.2513499).
- [56] R. M. Soldado and C. U. Almagro, “An overview of brdf models,” 2012. [Online]. Available: <https://api.semanticscholar.org/CorpusID:44900609>.
- [57] F. Fankhauser, J. A. Tyson, and J. Askari, “Satellite optical brightness,” *The Astronomical Journal*, vol. 166, no. 2, p. 59, Jul. 2023. DOI: [10.3847/1538-3881/ace047](https://doi.org/10.3847/1538-3881/ace047). [Online]. Available: <https://doi.org/10.3847%2F1538-3881%2Face047>.
- [58] P. Shirley and S. Marschner, *Fundamentals of Computer Graphics*, 3rd. USA: A. K. Peters, Ltd., 2009, ISBN: 1568814690.
- [59] W. Keeter, *Nasa 3d models*, <https://nasa3d.arc.nasa.gov/models/>, Accessed: 2022-04-27, Feb. 2013.
- [60] 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](https://doi.org/10.4103/0256-4602.107338).

- [61] S. Brabec, T. Annen, and H.-P. Seidel, “Practical shadow mapping,” *Journal of Graphics Tools*, vol. 7, no. 4, pp. 9–18, 2002.
- [62] R. Santamaria, *Raylib*, <https://www.raylib.com>, Accessed: 2022-04-28, Nov. 2021.
- [63] O. Montenbruck and E. Gill, *Satellite orbits: models, methods, and applications*. Springer Science & Business Media, 2012.
- [64] N. J. P. Laboratory, *Spice*, <https://naif.jpl.nasa.gov/naif/toolkit.html>, Accessed: 2023-10-24, 2023.
- [65] 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>.
- [66] L. T. Sharpe, A. Stockman, W. Jagla, and H. Jägle, “A luminous efficiency function, $V^*(\lambda)$, for daylight adaptation,” *Journal of Vision*, vol. 5, no. 11, pp. 3–3, Dec. 2005, ISSN: 1534-7362. doi: [10.1167/5.11.3](https://doi.org/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>.
- [67] G. M. Daniels, “A night sky model for satellite search systems.,” *Optical Engineering*, vol. 16, pp. 66–71, Feb. 1977.
- [68] K. A. Pickering, “The Southern Limits of the Ancient Star Catalog and the Commentary of Hipparchos,” *DIO*, vol. 12, pp. 3–27, Sep. 2002.
- [69] 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.
- [70] 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](https://doi.org/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>.
- [71] 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](https://doi.org/10.1051/0004-6361:20064992). arXiv: [astro-ph/0604128 \[astro-ph\]](https://arxiv.org/abs/astro-ph/0604128).

- [72] 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](https://doi.org/10.1051/0004-6361/202243940). arXiv: [2208.00211 \[astro-ph.GA\]](https://arxiv.org/abs/2208.00211).
- [73] J. C. Segovia, *Astroquery.gaia package*, <https://astroquery.readthedocs.io/en/latest/gaia/gaia.html>, Accessed: 2023-08-21, 2016.
- [74] F. E. Roach, “A Photometric Model of the Zodiacal Light,” *Astronomical Journal*, vol. 77, p. 887, Dec. 1972. doi: [10.1086/111363](https://doi.org/10.1086/111363).
- [75] M. Kaasalainen and J. Torppa, “Optimization methods for asteroid lightcurve inversion: I. shape determination,” *Icarus*, vol. 153, no. 1, pp. 24–36, 2001, ISSN: 0019-1035. doi: <https://doi.org/10.1006/icar.2001.6673>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0019103501966734>.
- [76] B. Horn, “Extended gaussian images,” *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1671–1686, 1984. doi: [10.1109/PROC.1984.13073](https://doi.org/10.1109/PROC.1984.13073).
- [77] 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](https://doi.org/10.1145/323233.323236). [Online]. Available: <https://doi.org/10.1145/323233.323236>.
- [78] C. L. Lawson and R. J. Hanson, “Solving least squares problems,” in *Classics in applied mathematics*, 1976. [Online]. Available: <https://api.semanticscholar.org/CorpusID:122862057>.
- [79] R. Bro and S. De Jong, “A fast non-negativity-constrained least squares algorithm,” *Journal of Chemometrics*, vol. 11, no. 5, pp. 393–401, 1997. doi: [https://doi.org/10.1002/\(SICI\)1099-128X\(199709/10\)11:5<393::AID-CEM483>3.0.CO;2-L](https://doi.org/10.1002/(SICI)1099-128X(199709/10)11:5<393::AID-CEM483>3.0.CO;2-L). [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-128X%28199709/10%2911%3A5%3C393%3A%3AAID-CEM483%3E3.0.CO%3B2-L>.
- [80] B. Keinert, M. Innmann, M. Sänger, and M. Stamminger, “Spherical fibonacci mapping,” *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–7, 2015.
- [81] E. W. Weisstein, *Sphere point picking*, <https://mathworld.wolfram.com/SpherePointPicking.html>, Accessed: 2022-07-03, 2002.

- [82] R. Gardner and P. Milanfar, “Reconstruction of convex bodies from brightness functions,” *Discrete and Computational Geometry*, vol. 29, pp. 279–303, Jul. 2003. DOI: [10.1007/s00454-002-0759-2](https://doi.org/10.1007/s00454-002-0759-2).
- [83] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*. Philadelphia, PA, USA: SIAM, 2000.
- [84] 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/>.
- [85] 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.
- [86] A. Bærentzen and H. Aanæs, “Generating signed distance fields from triangle meshes,” Jan. 2002.
- [87] D. Cohen-Or, D. Levin, and A. Solomovici, “Three-dimensional distance field metamorphosis,” *ACM Transactions on Graphics*, vol. 17, Feb. 2000. DOI: [10.1145/274363.274366](https://doi.org/10.1145/274363.274366).
- [88] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’87, New York, NY, USA: Association for Computing Machinery, 1987, pp. 163–169, ISBN: 0897912276. DOI: [10.1145/37401.37422](https://doi.org/10.1145/37401.37422). [Online]. Available: <https://doi.org/10.1145/37401.37422>.
- [89] W. Schroeder, R. Maynard, and B. Geveci, “Flying edges: A high-performance scalable isocontouring algorithm,” Oct. 2015. DOI: [10.13140/RG.2.1.3415.9609](https://doi.org/10.13140/RG.2.1.3415.9609).
- [90] S. M. Omohundro, *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.