

**Robust Shape and Pole Characterization of Small Bodies
using Infrared Images**

by

Koundinya Kuppa

B.S., Rutgers, The State University of New Jersey, 2014

M.S., The Pennsylvania State University, 2016

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Aerospace Engineering Sciences
2024

Committee Members:

Jay W. McMahon, Chair

Dr. Ann Dietrich

Prof. Daniel Scheeres

Prof. Marcus Holzinger

Prof. Paul Hayne

Kuppa, Koundinya (Ph.D., Aerospace Engineering Sciences)

Robust Shape and Pole Characterization of Small Bodies using Infrared Images

Thesis directed by Prof. Jay W. McMahon

Small body exploration provides three main benefits to our society: (1) as they are remnants of the early solar system, they are vital to our understanding of planetary formation, (2) they aid our planetary defense efforts, (3) they may be the key to the sustained exploration of our solar system. In order to facilitate the aforementioned benefits, characteristics of the body such as the orbit, size, shape, spin pole axis are necessary. Such characterization is performed by two main methods: ground based and spacecraft based. Ground based optical and radar telescopes are used to conduct large scale surveys of the solar system. These surveys produce a large catalog of small bodies with accurate orbit and size characterization but the spin pole axis and shape are often very poor. In order to produce improved characterization of any specific body of interest (*e.g* due to its high scientific value, high risk of collision etc.) spacecraft based is used. In this regime, a spacecraft with sensors such as cameras are sent to the body of interest and improved size, shape and pole axis information is obtained. The current state of the art algorithm to perform shape and pole axis estimation from spacecraft images is known as stereo-photoclinometry (SPC). SPC requires the use of high resolution optical images taken from specific illumination geometries to produce accurate shape and pole axis. The viewing geometry requirement in addition to the computational burden of the high resolution images prohibits the on-board, autonomous usage of SPC. Instead, a human-in-the-loop process is required in which a ground team processes the spacecraft images to produce the accurate spin pole axis and shape. Moreover, as optical images require an illumination source, SPC is not illumination robust.

In this thesis, we present algorithms that leverage infrared images from a spacecraft to estimate the pole axis and shape of a small body. In contrast with optical images, infrared images are more illumination robust as they detect re-radiated light in the infrared spectrum instead of

reflected light in the optical spectrum. The first algorithm obtains the pole estimate from a set of on-board infrared images. Due to the aforementioned advantage of infrared images, the proposed method is applicable in a vast majority observational geometries. In addition, the algorithm doesn't require a prior estimate. Once the pole axis is known, the silhouette (*i.e.* the edge between the body and the background) of space can be used to extract the shape of the body. While such Shape from Silhouette (SfS) methods can be applied to optical, infrared images offer the aforementioned robustness to illumination as well as more accurate characterization of the body's center of mass. In the proposed SfS method, an optimal sampling algorithm is outlined which extracts only the salient pixels from each image to ensure shape accuracy while reducing computational cost. While SfS algorithms are useful in generating initial shapes of small bodies, there are limitations. Some shape morphologies (*e.g.* craters) cannot be observed on the silhouettes and thus aren't accurately captured. While SPC can be used to refine the shape, we propose a new algorithm called stereo-thermoclinometry (STC) which uses measured temperatures from infrared images in conjunction with predicted temperatures from a thermophysical model (TPM) to refine the shape of an object. As direct surface temperature measurements are used, STC is able to characterize craters and other such concave shape morphologies. All algorithms presented in this thesis are tested on simulated infrared images and sensitivity analysis of relevant parameters is conducted to comprehensively characterize the algorithms. In addition to simulated data, some of the algorithms were tested on real infrared imagery when such data was available and applicable to the algorithm.

These algorithms enable more computationally efficient and illumination robust characterization of the spin pole axis and shape of small bodies. This enhances mission autonomy by reducing the need for ground teams for processing. More autonomous missions reduce overall costs and can allow for more bodies to be visited. By doing so, comprehensive characterization of more bodies is possible which leads to improved knowledge of planetary formation, more informed planetary defense capabilities, and may be the key to sustained exploration of the solar system through enabling in-situ resource utilization.

Dedication

To my mom and dad who mean everything to me.

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Jay McMahon, for guiding and mentoring me through my PhD. Jay's understanding and empathy have been extremely useful in handling the technical and mental challenges of the PhD. I would like to thank Dr. Ann Dietrich, for connecting me with Jay and subsequently serving as my Draper advisor. Her time, encouragement and advice has been invaluable to my success. Thank you to Prof. Marucs Holzinger, Prof. Dan Scheeres, and Prof. Paul Hayne for agreeing to serve on my committee and providing valuable feedback during my comprehensive exam which was vital in steering the research in the right direction. I would like to thank my fellow ORCCA labmates for being great friends and putting up with me for the last four years. I will cherish the encouragements, celebrations, commiserations, jokes, laughs, and fun we shared in and outside the lab. A special shoutout to my fellow navigation nerds, Dr. Jacopo Villa and Dr. Matthew Givens. I hope to continue to chat with you two about this nerdy stuff for eternity.

Lastly and certainly not the least, none of what I have done in my life has been possible without the unwavering support from my mom and dad. Their time, sacrifice and toil are the reasons for all of my successes as both as an engineer and as a person. From the bottom of my heart, thank you mom and dad.

Contents

Chapter

1	Introduction	1
1.1	Small Body Exploration Strategies	3
1.2	Sensors and Algorithms	5
1.2.1	Sensors	5
1.2.2	Algorithms	6
1.3	Thesis Statement	9
1.4	Thesis Outline	9
1.5	Publications	9
1.5.1	Journal Articles	9
1.5.2	Conference Papers	10
1.5.3	Invited Talks and Poster Presentations	11
1.5.4	Other Publications	11
2	Preliminaries	12
2.1	Frame Definitions	12
2.2	Thermal Modeling	13
2.2.1	Parametric Thermal Model	13
2.2.2	High Fidelity Thermal Model	16
2.3	Image Rendering	18

2.3.1	OpenGL Rendering	19
2.3.2	Rasterized Rendering	20
3	Spin Pole Estimation	22
3.1	Introduction and Background	22
3.1.1	Assumptions	22
3.2	Outline	23
3.3	Spin Pole Axis Estimation	23
3.3.1	Obtaining a Pole Estimate from Latitude Lines	24
3.3.2	Obtaining Latitude Lines from Images	27
3.4	Spin Direction Estimation	32
3.4.1	Appearance of the Spin Direction in the Image	32
3.4.2	Obtaining the Direction of Motion from Two Images	33
3.5	Spin Pole Estimation Algorithm	36
3.6	Results	37
3.6.1	Pole Estimation Method Applied to Simulated Bennu Images	38
3.6.2	Pole Estimation Method Applied to Simulated Itokawa Images	41
3.6.3	Pole Estimation Method Applied to Hayabusa2 Images of Ryugu	45
3.7	Algorithm Sensitivity Analysis	48
3.7.1	Algorithm Sensitivity to the Percentage of Spin Period Imaged	49
3.7.2	Algorithm Sensitivity to Image Spacing	50
3.8	Summary	52
4	Binary Orbit Pole Estimation	55
4.1	Introduction	55
4.2	Method	55
4.3	Results	58
4.3.1	Eccentricity vs. Inclination	59

4.3.2	Eccentricity vs. Right Ascension	59
4.3.3	Eccentricity vs. Argument of Periapsis	60
4.3.4	Eccentricity vs. Camera Latitude	62
4.3.5	Eccentricity vs. Camera Longitude	63
4.4	Summary	63
5	Shape from Silhouette	65
5.1	Introduction	65
5.1.1	Assumptions	65
5.2	Outline	66
5.3	Algorithm	66
5.3.1	Image Processing	67
5.3.2	Point Cloud Generation	71
5.3.3	Shape Reconstruction	77
5.3.4	SfS Algorithm	78
5.4	Results	80
5.4.1	Shape Models Derived from Synthetic Images	80
5.4.2	Shape Models using Images from Two Latitudes	84
5.4.3	Sensitivity to Pole Errors	86
5.4.4	Shape Model Derived from Hayabusa2 Images	91
5.5	Summary	92
6	Stereo Thermoclinometry	95
6.1	Introduction	95
6.1.1	Outline	96
6.2	Stereo-Thermoclinometry (STC)	96
6.2.1	Thermo-Physical Model (TPM)	96
6.2.2	Associating Pixel Temperatures to Facets	96

6.2.3	Thermoclinometry	99
6.2.4	Deriving Corresponding Vertices	101
6.2.5	STC Algorithm	105
6.3	STC Uncertainty	106
6.4	Results	110
6.4.1	Observational Viewpoint Sensitivity Analysis	113
6.4.2	Axial Tilt Sensitivity	115
6.4.3	Image Frequency Analysis	116
6.4.4	STC with Flyby Images	118
6.5	STC Uncertainty Monte Carlo	121
6.6	Summary	122
7	Conclusion	126
	Bibliography	129
	Appendix	
A	Determining Shadows using Ray-Tracing	140
B	Limb-Limb Intersection Procedure	142
C	Facet Normal Sensitivity to Vertices	143

Tables

Table

3.1	Camera Parameters for the Simulated Bennu and Itokawa Images	38
3.2	Pole Estimation Algorithm Parameters for Simulated Bennu Images	39
3.3	Pole Estimation Algorithm Parameters for Simulated Itokawa Images	42
3.4	Pole Estimation Algorithm Parameters For Hayabusa2 Images of Ryugu	46
3.5	Pole Estimation Results on Hayabusa2 Imagery	48
3.6	Adjusted cannysig Parameters for Image Spacing Study	51
4.1	Summary of parameters and results for the binary orbit pole study. For each case, orbital eccentricity and one other parameter are changed.	60
5.1	Statistics of shape models at 0° and 45° viewpoints. Rightmost column provides a reference to the figure showing the shape model performance	80
5.2	Statistics of shape models with pole errors. RMS and Errors shown in units of meters and as a percentage of the body radii. References to the shape models are given in the leftmost column for the corresponding small body	89
6.1	Thermal parameters for sphere with crater shape model (using Ryugu parameters) .	111
6.2	Camera and observation parameters	111

Figures

Figure

1.1	Distribution of small bodies in the solar system (Image Source: Wikimedia)	1
1.2	Discovery of Known Near Earth Asteroids (NEAs)(Image Source: Wikimedia)	2
1.3	Small bodies visited by spacecraft shown to scale (Image Source: The Planetary Society)	4
1.4	Comparison of Shape models derived using Ground based and spacecraft data	7
2.1	Graphical illustration of the geometry	13
2.2	Illustration of Facet Normal ($\hat{\mathbf{n}}_{facet}$), Incidence Angle (α), and Emission Angle (β) .	14
2.3	Simulated Bennu IR Images (at two different times)	16
2.4	Illustration of the surface boundary condition for a Thermo-Physical Model (TPM) .	18
2.5	Surface Temperatures as Modeled by a TPM	19
2.6	Rasterized Rendering Example Image (Image resolution is 500×500 pixels with a camera FOV of 5°)	21
3.1	Visualization of the Latitude Line Geometry	26
3.2	Procedure to Extract Latitude Lines from IR Images	28
3.3	Procedure to combine complementary streaks. Note that the blue circle denotes the match radius threshold (<code>matchThresh</code>)	31
3.4	Illustration of the In-Plane Ambiguity Resolution	32

3.5 Spin Direction Illustration (Fig. 3.5a shows an edge image extracted from stacking images)	33
3.6 Identifying Spin Direction (previous edge in green, current edge in pink)	35
3.7 Pole Estimate Results on Simulated Infrared Images of Bennu	39
3.8 Eccentricity Estimate Performance	40
3.9 Orientation Estimate Performance	41
3.10 Itokawa Edge Images Comparison (pink region shows used streaks, unused streaks in green)	42
3.11 Pole Estimate Results on Simulated Infrared Images of Itokawa	43
3.12 Itokawa Eccentricity Estimate Performance	44
3.13 Itokawa Orientation Estimate Performance	44
3.14 Hayabusa2 Infrared Images of Asteroid Ryugu	45
3.15 Obtaining Streaks from Hayabusa2 Images ((b)-(e) cropped to highlight asteroid) . .	47
3.16 Pole Error Histograms for Different Number of Bennu Images (Simulated Images) . .	51
3.17 Pole Error Histograms for Different Image Spacing on Simulated Bennu Images . . .	52
4.1 Geometry of a binary asteroid system modeled with Keplerian dynamics	56
4.2 Stacked Image aligned around the primary shows the projected orbit	57
4.3 Projected ellipse fit (in blue) with orientation ϕ , major axis $c\hat{\mathbf{a}}$ and minor axis $c\hat{\mathbf{b}}$. .	58
4.4 Orbit Pole Error across Different Orbital Eccentricities and Inclinations	61
4.5 Orbit Pole Error across Different Orbital Eccentricities and Right Ascensions	61
4.6 Orbit Pole Error across Different Orbital Eccentricities and Arguments of Periapsis .	62
4.7 Orbit Pole Error across Different Orbital Eccentricities and Observer Latitudes . . .	62
4.8 Orbit Pole Error across Different Orbital Eccentricities and Observer Longitudes . .	63
5.1 Shape from Silhouette (SfS) Algorithm Flowchart	67
5.2 Example infrared image of Ryugu, and the output of the binarization process	68

5.3	Illustration of the Moore neighborhood based Tracing Method. Each red box illustrates one iteration of the method. Bottom-left red box demonstrates Jacob's Stopping Criteria. Tracing Method highlighted for one pixel in the Inset	69
5.4	Illustration of the Optimal Sampling Algorithm	70
5.5	Silhouette pixel $\mathbf{u}$, line-of-sight ℓ , and line segment $\mathbf{L}$ with coordinates $\mathcal{B}\mathbf{r}_s$ and $\mathcal{B}\mathbf{r}_e$	72
5.6	Neighboring lines-of-sight $(\ell^{(i,1)}, \ell^{(i,2)}, \ell^{(i,3)})$ from image i , corresponding line segments $(\mathbf{L}^{(i,1)}, \mathbf{L}^{(i,2)}, \text{ and } \mathbf{L}^{(i,3)})$, limb sections $\mathbf{Q}^{(i,1)}, \mathbf{Q}^{(i,2)}$ (shaded green), and limb normals $\mathcal{B}\hat{\mathbf{n}}^{(i,1)}, \mathcal{B}\hat{\mathbf{n}}^{(i,2)}$	73
5.7	Demonstrating intersection $(\mathcal{B}\mathbf{i}_s, \mathcal{B}\mathbf{i}_e)$ between limbs from image i and image j ($i \neq j$)	74
5.8	Illustration of the trimming procedure	76
5.9	Illustration of Shape Point Extraction by overlapping trimmed segments $\mathbf{T}$	77
5.10	Characterization of Bennu shape model using images from 0° latitude viewpoint (color of points indicate the errors)	81
5.11	Characterization of Bennu shape model using images from 45° latitude viewpoint (color of points indicate the errors)	82
5.12	Characterization of Itokawa shape model using images from equatorial viewpoint (color of points indicate the errors)	83
5.13	Characterization of Itokawa shape model using images from 45° latitude viewpoint (color of points indicate the errors)	84
5.14	Characterization of 67P/C-G shape model using images from equatorial viewpoint (color of points indicate the errors)	84
5.15	Characterization of 67P/C-G shape model using images from 45° latitude viewpoint (color of points indicate the errors)	85
5.16	Characterization of Eros shape model using images from equatorial viewpoint (color of points indicate the errors). Absence of points indicate the limitation of the equatorial viewpoint.	85

5.17	Characterization of Eros shape model using images from 45° latitude viewpoint (color of points indicate the errors)	86
5.18	Shape models using images from 0° & 45° latitude viewpoints (color of points indicate the errors)	87
5.19	Comparison of performance across latitudes for 0°, 45°, and 0° & 45° cases	88
5.20	Performance of Shape Models with Pole Error	90
5.21	Characterization of Ryugu shape model using real images from Hayabusa2's TIR Camera (color of points indicate the errors)	91
6.1	Flowchart for the STC Algorithm. T_{pred} and T_{meas} are the predicted and measured surface temperatures respectively. $\hat{\mathbf{n}}_{est}$ are the estimated surface orientations. $\mathbf{v}_i$ are the updated shape vertices	97
6.2	Illustration of mapping pixel temperatures to facets	99
6.3	Vertices ($\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l$), Edges ($\mathbf{e}_{kj}, \mathbf{e}_{lk}, \mathbf{e}_{jl}$), True Normal ($\hat{\mathbf{n}}$), Estimated Normal ($\hat{\mathbf{n}}_{est,i}$) for a facet i	101
6.4	Quasi-Newton Update with a BFGS Update	105
6.5	True and initial shape models of a spherical body with a crater	110
6.6	Initial shape model error	111
6.7	Performance of the STC algorithm for a sphere shape model with a crater	112
6.8	Two images at different latitudes and longitudes showing the projection error. The true shape model is shown in black and the initial shape model is shown in white	113
6.9	RMS errors to illustrate viewpoint sensitivity at different iterations of STC (note that the colormap are different across the sub-figures)	114
6.10	Shape model RMS across axial tilts after 60 iterations of the STC algorithm	115
6.11	Shape models across different axial tilts between -9° and $+9^\circ$ after 60 iterations of the STC algorithm	117

6.12 RMS errors of shape model after 60 iterations of STC across at different sampling rates	118
6.13 Initial, true, and estimated incidence angles for facet 140 of the shape model. Also shown are the predicted and measured temperatures obtained from the TPM and IR images respectively	119
6.14 Geometry of the Flyby Scenario. Inbound section shown in red, outbound section shown in blue	120
6.15 Performance of the STC algorithm for a sphere shape model with a crater in a Flyby scenario illustrated in Fig. 6.14	120
6.16 Incidence angle Monte Carlos (1000 runs) for facet 1 of the shape model. Analytical uncertainty ($3\text{-}\sigma$) bounds shown in red	122
6.17 Facet normal Monte Carlos (1000 runs) for facet 1 of the shape model. Analytical uncertainty ($3\text{-}\sigma$) ellipse shown in red	123
6.18 Monte Carlo results (1000 runs) for vertex 1 of the shape model. Analytical uncertainty ($3\text{-}\sigma$) ellipse shown in red	123

Chapter 1

Introduction

The first asteroid (1) Ceres¹ was discovered in 1801 by Giuseppe Piazzi [32, 116]. Since then, many such small bodies have been discovered throughout the solar system (Fig. 1.1). As of 2024, approximately 1.4 million known asteroids and 4000 comets have been discovered².

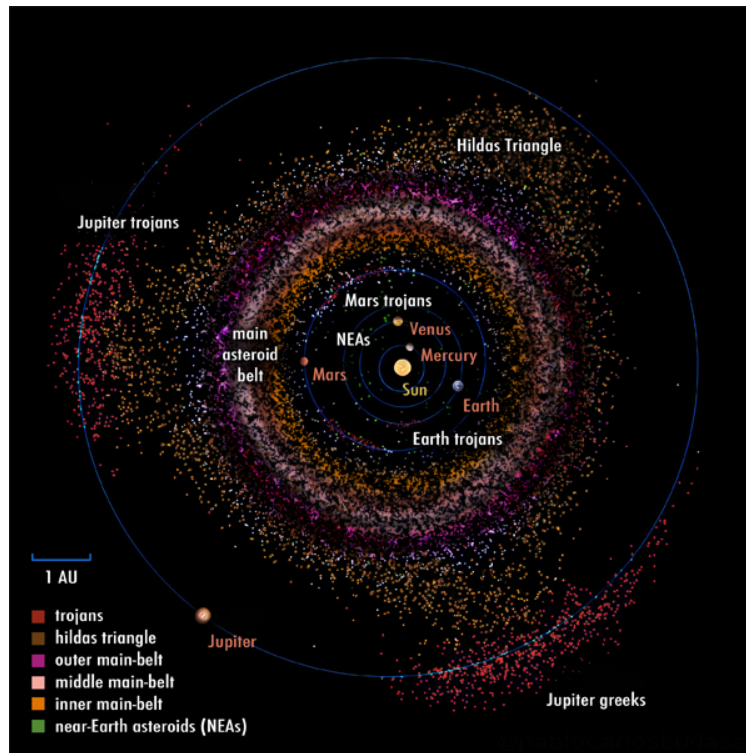


Figure 1.1: Distribution of small bodies in the solar system (Image Source: [Wikimedia](#))

These bodies exist in many regions throughout the solar system: the inner asteroid belt

¹ Ceres was later classified as a dwarf-planet

² <https://ssd.jpl.nasa.gov/>

between the orbits of Mars and Jupiter, the Kuiper Belt beyond orbit of Neptune, trailing and leading the orbit of Jupiter (the Trojan asteroids). Due to the gravitational influence of Earth, asteroids in the vicinity of the Earth (Near Earth Asteroids or NEAs) also exist. To date, more than 35,000 such NEAs have been discovered (Fig. 1.2). There are several benefits to studying

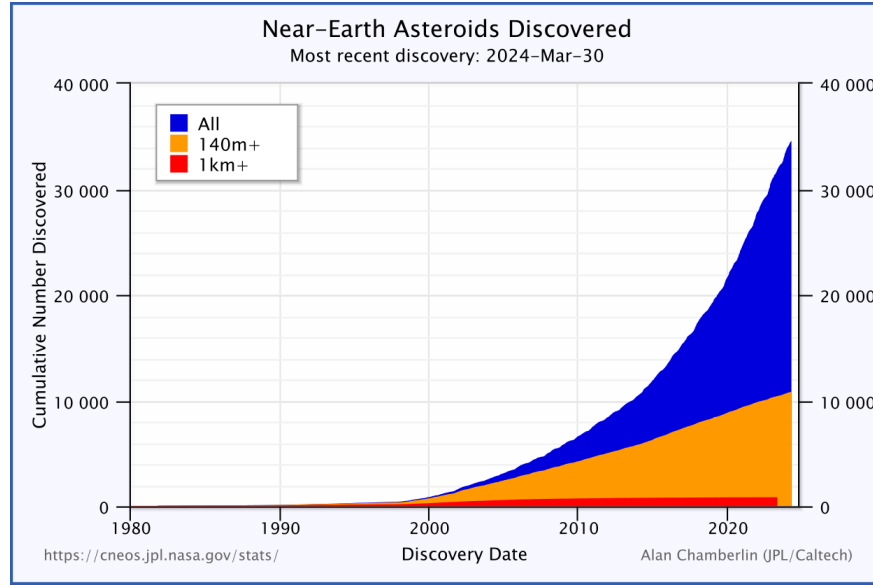


Figure 1.2: Discovery of Known Near Earth Asteroids (NEAs)(Image Source: [Wikimedia](#))

these bodies. These bodies are leftover from the planetary formation during the early solar system. As remnants of the this early time in the solar system, these bodies provide an insight into the formation of planetary bodies in our solar system [31, 83, 100]. Due to the gravitational influence of the Earth, some of the NEAs occasionally collide with our planet [28, 39, 108, 114]. Such collisions have severely negative impacts on human civilization, life, and infrastructure [111, 135]. The scale of the impact depends on the size, shape and composition of the colliding body. Thus, characterizing these objects aids in the monitoring and mitigation efforts of their impact. These bodies are composed of vital materials (water, metals etc) that could be serve as extraterrestrial sources for fuel for robotic or human exploration vehicles [13, 109].

1.1 Small Body Exploration Strategies

Ground based characterization of small bodies is performed using telescopes [26, 49, 61, 66, 118]. These telescopes collect photometric data by tracking a given body across the sky. This information is then used to infer the size, shape, composition, orbit etc. of these bodies. As they are on the ground, such telescopes are not constrained by power requirements and can effectively characterize many bodies over their operational lifetimes. This helps build a large catalog of small bodies for a relatively small cost. However, ground based sources do not always accurately characterize these bodies due to their observational geometry, atmospheric interferences, and limitations of the photometric data. Space based telescopes such as Gaia [84] offer an improvement over their ground-based counterparts by eliminating the atmospheric interferences but the other two issues remain.

To overcome limitations of Earth based (both ground and orbital) telescopes, robotic exploration of these bodies is an effective strategy to obtain more accurate data (especially about the shape and composition). Robotic exploration of small bodies can be divided into three mission types: (1) flybys, (2) rendezvous, and (3) sample return. To date, 18 small bodies have been visited by spacecraft (Fig. 1.3). Fly-by missions collect observations about one or more small bodies from a similar heliocentric orbit. Past examples of flybys include the Deep Space 1 spacecraft flyby of (4015) Wilson–Harrington³ and (9969) Braille [105]. Currently, the Lucy spacecraft [70] is operational and planning to flyby a set of Trojan asteroids over the next decade. The Emirates Mission to the Asteroid Belt (EMA) is scheduled to flyby six main belt asteroids [7, 99] and the DESTINY+ mission is planned to flyby the asteroid (3200) Phaeton [95, 112]. The advantage of such missions is that they are able to visit many target bodies because the fuel saved by *not* entering into orbit around any given body. This saved fuel can be spent to flyby other bodies. As these missions do not spend significant amount of time near any body, accurate characterization may not always be possible.

³ (4015) Wilson-Harrington is categorized both as an asteroid and a comet as it is active [52, 53]

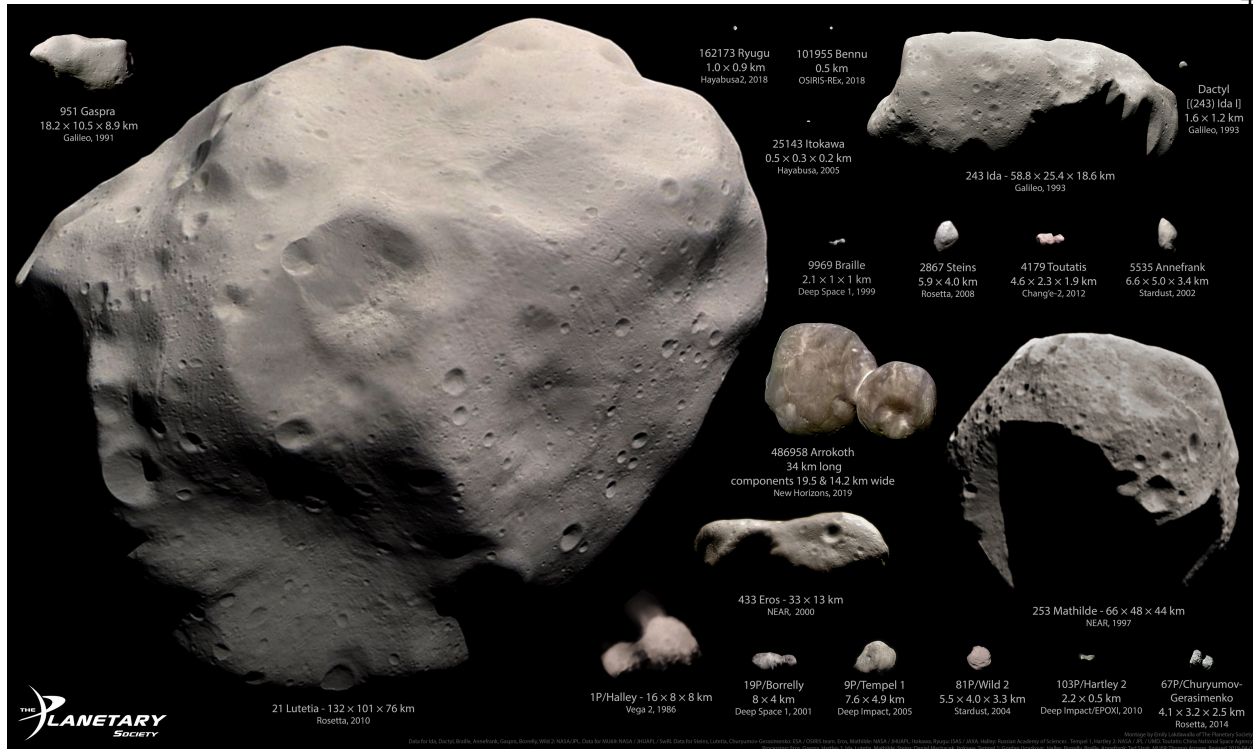


Figure 1.3: Small bodies visited by spacecraft shown to scale (Image Source: [The Planetary Society](https://www.planetary.org))

The second type of asteroid exploration, rendezvous, results in greater observational opportunities. In this type, a spacecraft enters into orbit around a small body and conducts extensive observations with a suite of on-board instrumentation (cameras, spectrographs, radars etc.). With the increased time for an observational campaign, a thorough characterization is possible. Data such as images and spectral signatures are used to derive the shape, composition, gravity of the small body. A discussion of such sensors and the associated algorithms is discussed in Section 1.2. The Dawn mission [131] to Vesta and Ceres, the Psyche mission [72] to the asteroid (16) Psyche, the Rosetta Mission [46] to the comet 67P/Churyumov–Gerasimenko, the NEAR Shoemaker mission to asteroid (433) Eros [104], the proposed Janus Mission [113], the EMA mission [104] to asteroid (269) Justitia are a few examples of such rendezvous missions.

All exploration strategies previously discussed have utilized indirect measurements of the small body. To overcome the limitations of these indirect measurements, sample collection has been proposed. In this mission type, a spacecraft collects a sample from the surface is collected for

further analysis. In some cases, these samples are returned to Earth for a more detailed analysis. This mission type is advantageous over the other types as it allows for extensive analysis of the material. Missions such as OSIRIS-REx to asteroid Bennu [67], Hayabusa2 to asteroid Ryugu [137] have successfully collected samples [69, 140] from their respective asteroids.

1.2 Sensors and Algorithms

1.2.1 Sensors

Depending of the exploration strategy, different instruments and sensors are used to gather the necessary data that improves our understanding of the small body. In ground/space based surveys such as LINEAR or Gaia, optical (or sometimes infrared) telescopes are used to collect information from their respective Earth-proximity locations.

On-board a spacecraft, optical and infrared cameras, LIDARs and spectrometers are often used to obtain information. Such space rated sensors are subject to SWaP (Size Weight and Power) constraints as well as solar and cosmic radiation sources. Optical cameras produce images which provide observations of the small body in the optical spectrum ($\sim 380 - 750$ nm). These sensors are very low cost as they benefit from their ubiquity in terrestrial usage. Such cameras require calibration procedures on-ground and in space to enable their usage [19, 27, 47]. These passive sensors work by detecting the reflected light from the body using semiconductors sensitive to the optical spectrum. As a result, they require the body to be illuminated by the sun in order to be effective. This imposes constraints regarding the trajectory of the spacecraft relative to the small body.

A LIDAR (Light Detection and Ranging) is an active sensor that consists of a light emitter and a light detector. Based on the time difference between the emitted and detected light, information about the surface can be extracted. As LIDARs emit their own light, they are able to observe regions which are otherwise shadowed in the optical spectrum. However, this significantly increases the power requirements. In addition, LIDARs only work when they are close to the surface and

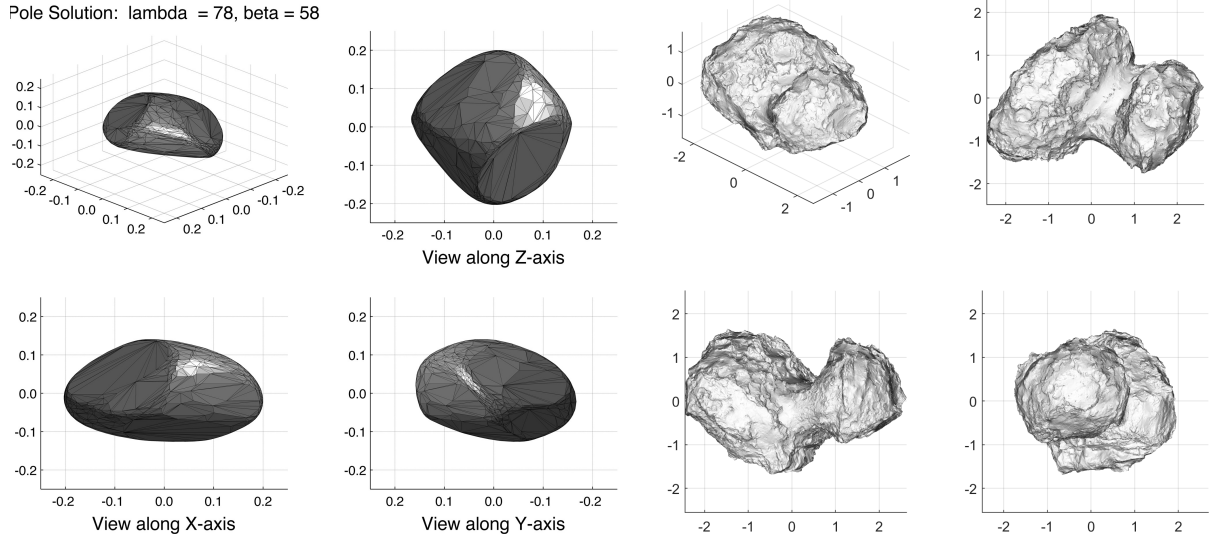
thus aren't suitable in flyby missions.

An infrared camera (bolometer) is a sensor that detects the blackbody radiation [89]. As all objects above absolute zero emit blackbody radiation, these sensors are able to image surfaces even when they are not directly illuminated. As they do not require any emitted light, their power requirements are lower than those of LIDARs. However, as they are not as widespread as optical cameras, the image resolution of infrared cameras is comparatively lower. Like their optical counterparts, a calibration process [10] is necessary to extract the relevant information from the raw output of the sensor. Surface temperature information derived from a calibrated infrared camera is used to infer thermal properties of the material [92]. Infrared cameras combine some of the advantages of optical cameras and LIDARs while mitigating their disadvantages.

1.2.2 Algorithms

In this section, an overview of the algorithms that utilize the data collected from aforementioned sensors (Section 1.2.1) to extract the spin pole orientation and shape of the body is presented. Photometric data from telescopes (both ground and space based) use a method called *lightcurve-inversion* [59, 60] to concurrently derive estimates of both the spin pole and the associated shape. Due to the size of the bodies and their distance from all Earth based telescopes, they do not appear as a resolved object (*i.e.* span more than a pixel on the sensor). Instead, a body's brightness over time (called a *lightcurve*) as it spins is observed from photometric data. This one-dimensional lightcurve is correlated with the body's shape and spin orientation. The aforementioned lightcurve inversion identifies the shape and pole orientation that best describes the observed lightcurve. Since the shape model is three-dimensional and the lightcurve is one-dimensional, the inverse mapping (*i.e.* from the lightcurve to the model) is not homeomorphic. This implies that many shape models (and associated pole axes) can produce the same lightcurve. As a result of this, lightcurve derived shape models may not accurately represent the true shapes and spin pole orientations of the body. Moreover, lightcurve inversion method often is limited by the fixed observational geometry between the body, Earth, and the Sun. Thus, the accuracy of the resultant shape models can be

poor. The limitations of the lightcurve inversion method is evident by comparing the ground based and spacecraft derived shape models Fig. 1.4. The associated spin pole estimates can also have a large uncertainty [48].



(a) Model of comet 67P/C-G derived using lightcurve inversion. (Image from Ref. [74]) (b) Model of comet 67P/C-G derived using spacecraft images [22]

Figure 1.4: Comparison of Shape models derived using Ground based and spacecraft data

This lightcurve inversion technique is also utilized on data gathered from spacecraft optical cameras when the spacecraft is far from the target body such that it remains unresolved. Once it is resolved (spans more than one pixel), other techniques can be used to derive more accurate shapes. Limb-based shape modeling techniques uses the visible edges of the body (against the background of space) in a set of optical images to infer its shape. Such so called shape-from-silhouette (SfS) methods [14, 15, 23, 87, 124] are able to capture certain shape concavities better than the lightcurve inversion method due to two reasons: and (1) two-dimensional nature of silhouettes (2) variable observational geometry. As the silhouettes are two-dimensional (as opposed to one-dimensional lightcurves), SfS methods are more homeomorphic than lightcurve inversion methods. Additionally, the observational geometry of a spacecraft (relative to the body) can be changed to provide different views of the body (unlike the relatively fixed geometry from Earth). Thus, SfS derived

shape models are more accurate. However, certain shape concavities such as craters are unable to be accurately characterized as they cannot be extracted from the silhouettes. If the images are taken from close proximity of the small body, a technique known as Stereo-photoclinometry (SPC) [43] is used to derive the shape of the body. SPC uses features on the body in conjunction with albedo modeling to estimate the shape of the body using several high resolution optical images taken from different illumination and observational geometries. While this technique has been used to produce accurate shape models [22, 35, 40, 41, 42, 96, 98], it requires an extensive mapping campaign to collect the images from the optimal illumination and observational geometries. The use of high-resolution imagery imposes a computational burden and thus prevents on-board autonomous shape characterization. SPC constructs shapes by iteratively refining coarser shape models which requires rendering capabilities [96] that are computationally prohibitive. Moreover, tracking features between images (*i.e.* data association) is significantly affected by the variations in lighting conditions. As SPC requires specific lighting conditions, manual association of the appropriate features between the images is performed. These computational burdens and manual aspects of SPC prevents its on-board and autonomous usage.

Unlike cameras, LIDARs can derive both the direction and distance to the shape. Thus, they have been used to derive the shape of and navigate relative to a small body [1, 80, 115]. However, the reflective properties of the surface can impact the quality of the data and the resultant shape. The OLA (OSIRIS-REx Laser Altimeter) data of asteroid Bennu was combined with SPC to improve the overall shape models [42]. As the altimeter provides an independent and direct range measurement, it improves the overall shape especially in regions not captured by SPC images (*e.g.* shadowed regions).

The data from thermal cameras is useful in determining features across the surface of an asteroid as well as estimating the thermal properties of the surface [92]. The temperature variations are indicative of the porosity of the surface. Algorithms have been proposed in literature to utilize infrared images for spacecraft navigation [30, 29, 142]. Authors in Ref. [29] propose a fusion of infrared and optical images whereas Ref. [142] propose its use for Active Debris Removal (ADR).

Optical feature descriptors have been tested on infrared images [54, 107] and infrared specific feature descriptors have also been proposed [5, 144]. As of the writing of this thesis, no literature on the use of infrared images for small body shape reconstruction and navigation has been proposed.

1.3 Thesis Statement

This thesis details new algorithms that leverage on-board infrared imagery to produce estimates of the pole axis and shape model of small bodies. Such algorithms are useful for light-robust proximity operations and scientific data collection around these bodies.

1.4 Thesis Outline

The thesis is outlined as follows. Chapter 2 outlines frame definitions, thermal models and image rendering techniques used in this document. Chapter 3 outlines an algorithm that estimates the pole axis from a set of infrared images. Chapter 4 outlines a related algorithm to estimate the orbit pole axis of a binary asteroid system. Chapter 5 outlines an algorithm to produce an initial shape of a small body from the observed silhouettes in a set of infrared images. Chapter 6 outlines an algorithm to refine the shape model based in the observed temperatures from infrared images. Finally, Chapter 7 provides concluding remarks and future research directions related to the thesis.

1.5 Publications

The following is a list of published and in-preparation journal articles as well as conference presentations related to this thesis work.

1.5.1 Journal Articles

- J-1 K. Kupa, J. W. McMahon, and A. B. Dietrich. Initial pole axis and spin direction estimation of asteroids using infrared imagery. **Journal of Guidance, Control, and Dynamics**, 47(6):1055–1071, 2024 (Chapter 3)

J-2 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Autonomous shape modeling of small bodies using infrared image silhouettes. **Acta Astronautica**, 2024 (*submitted*) (Chapter 5)

J-3 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Stereo thermoclinometry: A method to characterize small body shapes using infrared images. 2024 (*in prep*) (Chapter 6)

1.5.2 Conference Papers

C-1 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Stereo thermoclinometry (stc): A method to characterize small body shapes using infrared images. In **AAS/AIAA Astrodynamics Specialist Conference**, Broomfield, CO, USA, August 2024 (Chapter 6)

C-2 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Light-robust and autonomous spin pole estimation and shape modeling of asteroids using infrared imagery. In **2024 AAS GN&C Conference**, Breckenridge, CO, USA, February 2024 (Chapter 5)

C-3 J. Villa, **K. Kuppa**, J. W. McMahon, and J. Knittel. Silhouette-based pole estimation and shape reconstruction of asteroid 269 justitia. In **2024 AAS GN&C Conference**, Breckenridge, CO, USA, February 2024 (Chapter 5)

C-4 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Binary asteroid orbit pole estimation from spacecraft imagery. In **2023 Asteroids, Comets, Meteors Conference**, Flagstaff, AZ, USA, June 2023 (Chapter 4)

C-5 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Initial pole estimation using infrared imagery. In **2022 Space Imaging Workshop**, Atlanta, GA, USA, October 2022 (Chapter 3)

C-6 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Initial pole estimation of small bodies on approach using infrared imagery. In **2022 AAS/AIAA Astrodynamics Specialist Conference**, Charlotte, NC, USA, August 2022 (Chapter 3)

1.5.3 Invited Talks and Poster Presentations

T-1 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Stereo thermoclinometry (stc): A method to characterize asteroid shapes using infrared images. In **2024 Draper Research Symposium**, April 2023 (Chapter [6](#))

T-2 **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Towards autonomous pole estimation of small bodies. In **2023 Draper Research Symposium**, April 2023 (Chapter [3](#))

1.5.4 Other Publications

P-1 M. W. Givens, **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Autonomous cislunar navigation using a visual direction of motion measurement and a gaussian mixture model filter. In **AAS/AIAA Astrodynamics Specialist Conference**, Big Sky, MT, USA, August 2023

Chapter 2

Preliminaries

2.1 Frame Definitions

Three right-handed coordinate frames are used throughout this thesis: an inertial frame $\mathcal{I}$, a body frame $\mathcal{B}$, and a camera frame $\mathcal{C}$. An illustration of these frames is given in Fig. 2.1. The body frame $({}_B\hat{\mathbf{x}}, {}_B\hat{\mathbf{y}}, {}_B\hat{\mathbf{z}})$ is a rotating frame centered at the small body center of mass (CoM) with the ${}_B\hat{\mathbf{z}}$ aligned with the pole orientation ${}_I\hat{\mathbf{p}}$ (the axis about which the body spins). The inertial frame $({}_I\hat{\mathbf{x}}, {}_I\hat{\mathbf{y}}, {}_I\hat{\mathbf{z}})$ is also centered at the small body CoM with ${}_I\hat{\mathbf{z}}$ aligned with the ${}_B\hat{\mathbf{z}}$ and the pole axis. Over time, the body frame $\mathcal{B}$ rotates with the body as the inertial frame $\mathcal{I}$ remains stationary. Thus, the transformation between the inertial $\mathcal{I}$ and body frames $\mathcal{B}$ is a rotation about the z-axis B_R .

An observer with a camera is at ${}_I\mathbf{r}$ from the center of the body. A camera frame $({}_C\hat{\mathbf{x}}, {}_C\hat{\mathbf{y}}, {}_C\hat{\mathbf{z}})$ is centered at ${}_I\mathbf{r}$ with the ${}_C\hat{\mathbf{x}}$ and ${}_C\hat{\mathbf{y}}$ aligned with the horizontal and vertical image axes of the image and the boresight axis ${}_C\hat{\mathbf{z}}$ complete the right-handed triad. A given point on the body ${}_B\mathbf{s}$ projects to the point ${}_C\mathbf{u}$ on the image plane as shown in Fig. 2.1. The angles α_{cam} and ψ_{cam} define the camera vector in the body frame $\mathcal{B}$ and are the body relative longitude and latitude respectively. The angle from the equatorial plane of the body (plane spanned by ${}_B\hat{\mathbf{x}}, {}_B\hat{\mathbf{y}}$) to the direction of the sun ${}_I\hat{\mathbf{u}}_{sun}$ is known as the axial tilt γ . The angle between ${}_I\hat{\mathbf{u}}_{sun}$ and ${}_I\mathbf{r}$, is known as the phase angle ϕ (Eq. (2.1)).

$$\phi = \cos^{-1}({}_I\hat{\mathbf{r}}_{cam} \cdot {}_I\hat{\mathbf{u}}_{sun}) \quad (2.1)$$

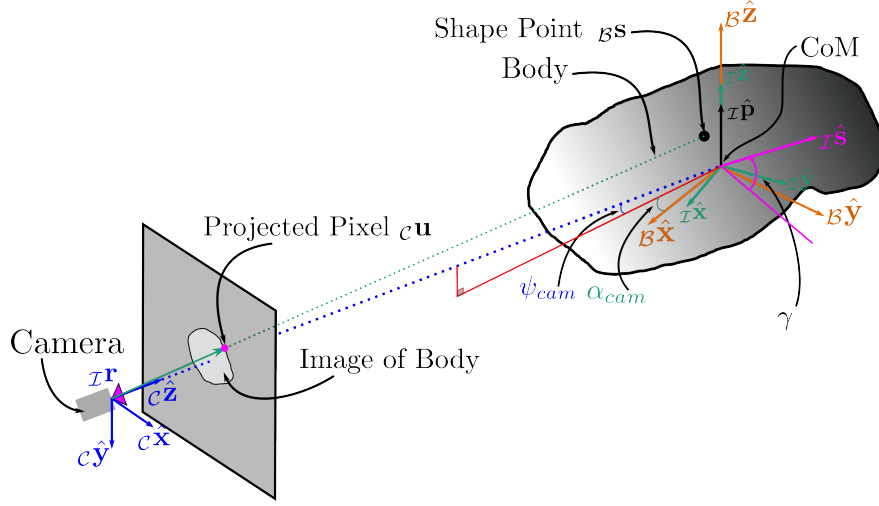


Figure 2.1: Graphical illustration of the geometry

2.2 Thermal Modeling

A thermal model is necessary to simulate the temperature of a given object (*e.g.* asteroid, comet, etc.). These temperatures are then used to render infrared images (IR). Thermal models either directly integrate the heat-equation to accurately simulate the temperature [30, 33, 76, 79] or use a parametric equation to approximate the temperature [65]. Both of these approaches are employed in this dissertation with a novel parametric model outlined in Section 2.2.1 and a thermophysical model outlined in Section 2.2.2.

2.2.1 Parametric Thermal Model

This thermal model is designed to simulate the macroscopic phenomenon that appear in infrared imagery. These phenomenon are primarily dependent on the geometry between the asteroid, Sun, and camera. This model produces images that can be used to rigorously test an infrared image based algorithm without incurring the computational burden of higher fidelity thermal models. Delbo et al. [33] provide a survey of thermal models for planetary surfaces and propose one such thermal model. Recent efforts have been made in thermal image simulation to facilitate validation

and testing of navigation algorithms [30, 79].

For a given facet (shown in Fig. 2.2), the angle between the facet normal $\hat{\mathbf{n}}_{facet}$ and the solar direction ${}^B\hat{\mathbf{u}}_{sun}$ (shown in Fig. 2.1) is known as the incidence angle α . This angle determines whether the facet is in an illuminated or shadowed state (in the absence of occlusions). These two states correspond to the warming and cooling modes of the algorithm, respectively. Occlusion of a given facet by other facets is considered in the model. When a facet is occluded, it is considered to be in a shadowed state and thus the cooling mode is active. The observed intensity is then computed based on the emission angle β which is the angle between the facet normal $\hat{\mathbf{n}}_{facet}$ and the direction from the facet to the observer/camera ${}^B\hat{\mathbf{r}}_{cam}$. Fig. 2.2 contains a graphical illustration of the incidence and emission angles.

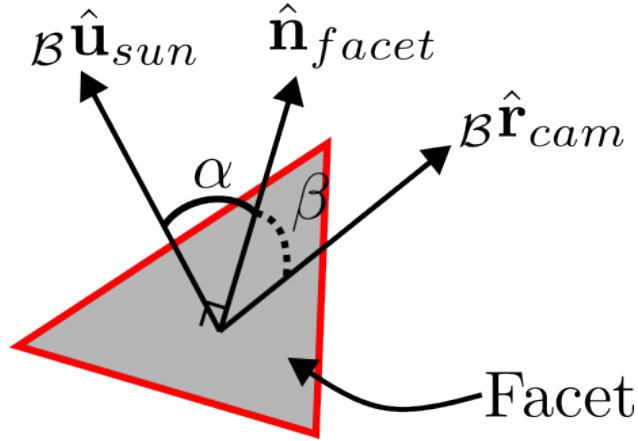


Figure 2.2: Illustration of Facet Normal ($\hat{\mathbf{n}}_{facet}$), Incidence Angle (α), and Emission Angle (β)

The thermal model is summarized in Algorithm 1. The warming mode of the model is active for a facet when it is illuminated (line 7 of Algorithm 1). The temperature increase is modeled as an exponential growth function (Eq. (2.2)). On the other hand, when a facet is shadowed, the cooling mode is active (line 10 of Algorithm 1). In this mode, the temperature decrease is modeled

as an exponential decay function (Eq. (2.3)).

$$T(t_k) = T(t_{k-1}) (1 - e^{-\cos \alpha r_w t_k}) \quad (\text{warming model}) \quad (2.2)$$

$$T(t_k) = T(t_{k-1}) e^{-r_c t_k} \quad (\text{cooling model}) \quad (2.3)$$

where

$T(t_{k-1})$: facet temperature at previous time step

$T(t_k)$: facet temperature at current time step

r_w : warming rate

r_c : cooling rate

Since this simplified model is operated on each facet independently, thermal flux between neighboring facets is not considered. The warming (r_w) and cooling (r_c) rates dictate how quickly the temperature increases or decreases. These parameters are to be set by a user based on the material properties of the asteroid. This model produces a temperature value for each facet. The infrared intensity (as observed by a camera) is computed using the emission angle β as described by Eq. (2.4).

$$I_{obs}(t_k) = T(t_k) \cos \beta \in [0, 1] \quad (2.4)$$

Therefore, the highest intensity is observed when the emission angle β is 0° which occurs when the camera direction (from the facet) is aligned with the normal direction of the facet. Conversely, the lowest intensity is observed at a β of 90° .

The model is applied to a 200,000 facet shape model of the asteroid Bennu [16]. Fig. 2.3 shows modeled intensities at two different times as the body spins about its pole axis. Owing to the reflected intensity (Eq. (2.4)) and the globally convex shape, the silhouette of the body always appears cooler in comparison to the rest of the body as the emission angles of facets close to the edge are closer to 90° .

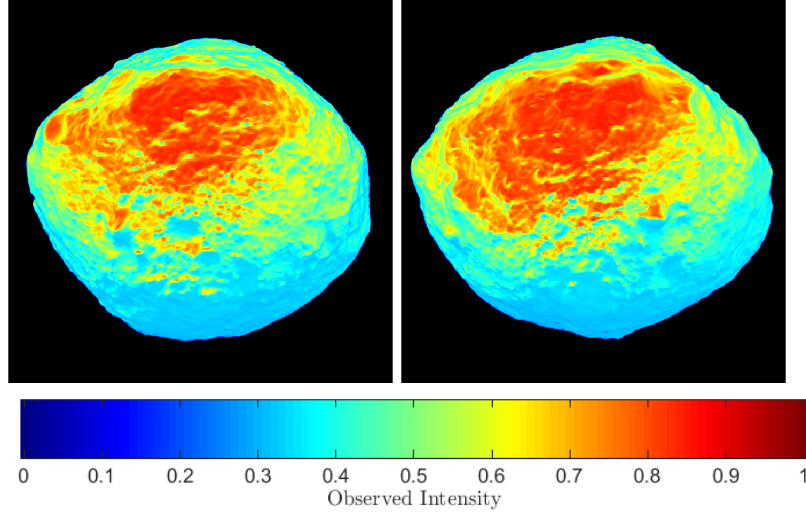


Figure 2.3: Simulated Bennu IR Images (at two different times)

Algorithm 1 Parametric Thermal Model

Given: r_w, r_c, F, V

Result: T

```

1: for  $k = 1 : nTime$  do
2:   % Compute shadows at current time
3:    $s(:, k) = \text{getShadows}(F, V, \hat{\mathbf{n}}, \hat{\mathbf{u}}_{sun, k})$ 
4:   for  $i = 1 : \text{numFacets}$  do
5:     if  $s(i, k) > 0 \ \& \ s(i, k-1) > 0$  then
6:        $t(k, i) \leftarrow t(k-1, i) + \Delta t$ 
7:        $T(k, i) \leftarrow (1 - T(k-1, i))e^{-\hat{\mathbf{n}}_i \cdot \hat{\mathbf{u}}_{sun} r_w t(k, i)}$ 
8:     else if  $s(i, k) \leq 0 \ \& \ s(i, k-1) \leq 0$  then
9:        $t(k, i) \leftarrow t(k-1, i) + \Delta t$ 
10:       $T(k, i) \leftarrow T(k-1, i)e^{-r_c t(k, i)}$ 
11:     else if  $s(i, k) \leq 0 \ \& \ s(i, k-1) > 0$  then
12:        $t(i, k) \leftarrow 0$ 
13:        $T(k, i) \leftarrow T(k-1, i)$ 
14:     else if  $s(i, k) > 0 \ \& \ s(i, k-1) \leq 0$  then
15:        $t(i, k) \leftarrow 0$ 
16:        $T(k, i) \leftarrow T(k-1, i)$ 
17:     end if
18:   end for
19: end for
```

▷ Temperature history for each facet
 ▷ Loop through time
 ▷ Appendix A
 ▷ Loop through facets
 ▷ Facet is currently lit and previously lit
 ▷ Update time
 ▷ Apply Warming Model
 ▷ Facet is currently shadowed and previously shadowed
 ▷ Update time
 ▷ Apply Cooling Model
 ▷ Facet is currently shadowed and previously lit
 ▷ Reset time for this facet
 ▷ Temperature Doesn't Change
 ▷ Facet is currently lit and previously shadowed
 ▷ Reset time for this facet
 ▷ Temperature Doesn't Change

2.2.2 High Fidelity Thermal Model

In some cases, accurate modeling of the surface temperatures are necessary (such as for Chapter 6). In such situations, the parametric model illustrated in Section 2.2.1 is inadequate. In this section, an overview of a high fidelity modeling scheme is presented which is used to accurately simulate the surface temperatures of a given body.

Given the sun direction, an initial shape model, and physical properties of a small body, a Thermo-Physical Model (TPM) [62, 76, 117] simulates the thermodynamics of the body. Such

TPMs model the temperatures of the body over time as different parts of the surface get illuminated by the sun. In general, these TPMs numerically integrate the heat equation (Eq. (2.5)) for a given surface element (*e.g.* a facet of a shape model). This form of the heat equation assumes that the thermal conductivity κ is not a function of depth z .

$$\frac{\partial T}{\partial t} = \frac{\kappa}{\rho C_p} \frac{\partial^2 T}{\partial z^2} \quad (2.5)$$

In Eq. (2.5), ρ is the density and C_p is the heat capacity. Two boundary conditions are required to solve this partial differential equation: (1) an internal boundary condition (Eq. (2.6)) that dictates a depth at which the temperature stops changing and (2) a surface boundary condition (Eq. (2.7)) that dictates the energy balance at the surface of the facet element.

$$\left. \frac{\partial T}{\partial z} \right|_{z \gg l_s} = 0 \text{ (Internal BC)} \quad (2.6)$$

$$F_{sun} + F_{cond} - F_{rad} = 0 \text{ (Surface BC)} \quad (2.7)$$

$$l_s = \sqrt{\frac{\kappa P_{rot}}{2\pi\rho C_p}} \quad (2.8)$$

$$F_{sun} = \frac{S_{\odot}(1-A)}{R_{AU}^2} \cos(\alpha)(1-s) \quad (2.9)$$

$$F_{cond} = \kappa \left. \frac{dT}{dz} \right|_{surf} \quad (2.10)$$

$$F_{rad} = \epsilon_B \sigma_{SB} T_{surf}^4 \quad (2.11)$$

The internal boundary condition Eq. (2.6) is dictated by the *skin depth* parameter l_s . At a depth z of several multiples of the parameter l_s , the temperature stops changing (*i.e.* the thermal gradient is zero). An illustration of the surface boundary condition (Eq. (2.7)) for a facet is shown in Fig. 2.4. This boundary condition enforces the first law of thermodynamics *i.e.* energy radiated from the sun F_{sun} plus the energy conducted through the surface F_{cond} be equal to the energy radiated to space F_{rad} (*i.e.* the energy is conserved). The angle between the sun direction ${}_B\hat{\mathbf{u}}_{sun}$ and the facet normal ${}_B\hat{\mathbf{n}}_{facet}$ is defined as the incidence angle α and determines how much energy from the sun is input to the surface.

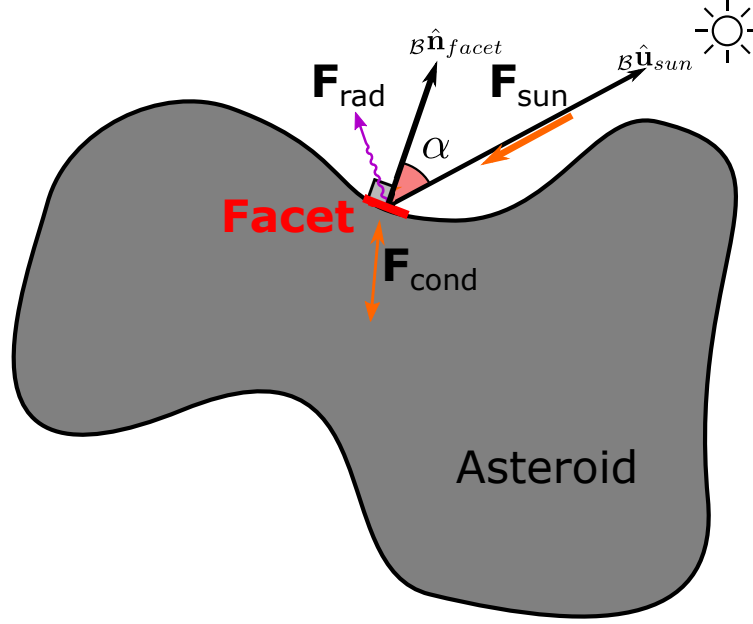


Figure 2.4: Illustration of the surface boundary condition for a Thermo-Physical Model (TPM)

TPMs discretize the surface into several nodes along the depth z direction for a set multiples of the skin depth l_s . Below this set depth, the temperature gradient is zero (the internal boundary condition, Eq. (2.6)) and then use a finite difference scheme to propagate each node forward in time. In this work, the TPM described in Ref. [62] is used¹. This TPM is applied to a shape model of the asteroid Ryugu and the resultant surface temperatures for all facets at one time are shown in Fig. 2.5a. Fig. 2.5b shows the temperature evolution for one facet over one spin period of the asteroid.

2.3 Image Rendering

The methods outlined in Section 2.2 are used to simulate the temperatures of a given small body. In this section, two approaches to generating synthetic infrared images (IR) are presented. As with the thermal models, both these approaches are used in this dissertation.

The simplified rendering scheme(Section 2.3.1) is sufficient for rendering silhouettes of the body as well as simulating the macroscopic thermal variation simulated by the parametric ther-

¹ This TPM is available to be used through the [Astroshaper](#) toolkit

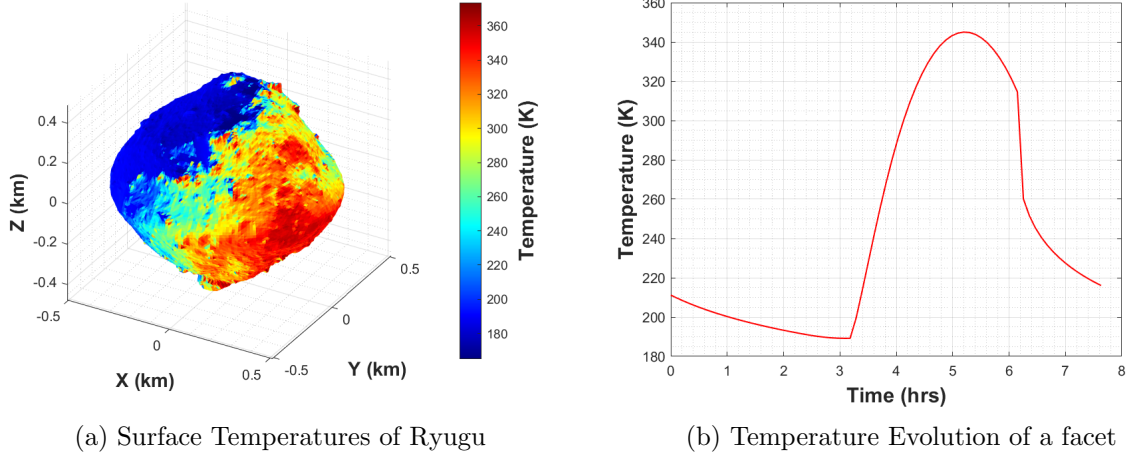


Figure 2.5: Surface Temperatures as Modeled by a TPM

mal model (Section 2.2.1) without incurring the computational cost of the rasterization process (Section 2.3.2). Thus, it is used to simulate images to test algorithms outlined in Chapters 3 to 5 where the additional fidelity of the rasterization process is unnecessary. As the OpenGL rendering is available through many rendering software (MATLAB, Blender etc.), its usage removes development and validation time of a higher fidelity renderer when such fidelity isn't necessary. However, for the algorithm outlined in Chapter 6, the additional fidelity is necessary to accurately simulate the expected temperature measurements from an infrared camera.

2.3.1 OpenGL Rendering

Using either thermal modeling procedure, a temperature for each facet of a shape model is determined. In MATLAB, this shape model is plotted and the facets shaded with the temperature (at a given time). MATLAB's OpenGL based rendering can then be used to generate simulated images of a specific image resolution and with a given camera field-of-view. Examples of such images are shown in Fig. 2.3.

2.3.2 Rasterized Rendering

The OpenGL rendering approach has a few limitations: (1) image resolution is tied to screen resolution and thus high resolutions cannot be easily rendered, (2) cameras with different horizontal and vertical field-of-views cannot accurately be simulated and (3) the precision in the image intensities (*i.e* temperatures) are limited by a 8-bit unsigned integer. To overcome these issues, a different approach is taken in which the facets (and corresponding vertices) of a shape model that are visible to the given camera location are determined using a ray-tracing algorithm (Appendix A). For each facet deemed to be in view, its center coordinates ${}_B\mathbf{f}_c$ are computed in the body frame as the average of its constituent vertices (${}_B\mathbf{v}_j, {}_B\mathbf{v}_k, {}_B\mathbf{v}_l$).

$${}_B\mathbf{f}_c = \frac{{}_B\mathbf{v}_j + {}_B\mathbf{v}_k + {}_B\mathbf{v}_l}{3} \quad (2.12)$$

The center coordinates of the facets in view are projected into the image plane as follows. The relative vector between the camera location in the body frame ${}_B\mathbf{r}$ and the facet center ${}_B\mathbf{f}_c$:

$${}_B\mathbf{d} = {}_B\mathbf{f}_c - {}_B\mathbf{r} \quad (2.13)$$

Using the body frame to camera frame transformation matrix ${}_B^C\mathbf{R}$, ${}_B\mathbf{f}_c$ is expressed in the camera frame:

$${}_C\mathbf{f}_c = {}_B^C\mathbf{R} {}_B\mathbf{f}_c \quad (2.14)$$

Given focal length f , the principal points of the camera s_x and s_y , and the pixel pitch μ_x and μ_y , the camera calibration matrix K is represented as:

$$K = \begin{bmatrix} d_x & 0 & \mu_x \\ 0 & d_y & \mu_y \end{bmatrix} \quad (2.15)$$

$$d_x = \frac{f}{\mu_x} \quad (2.16)$$

$$d_y = \frac{f}{\mu_y} \quad (2.17)$$

Pixel coordinates of ${}_C\mathbf{d}$ are computed using Eq. (2.15):

$$\mathbf{u}_d = K {}_C\mathbf{f}_c \quad (2.18)$$

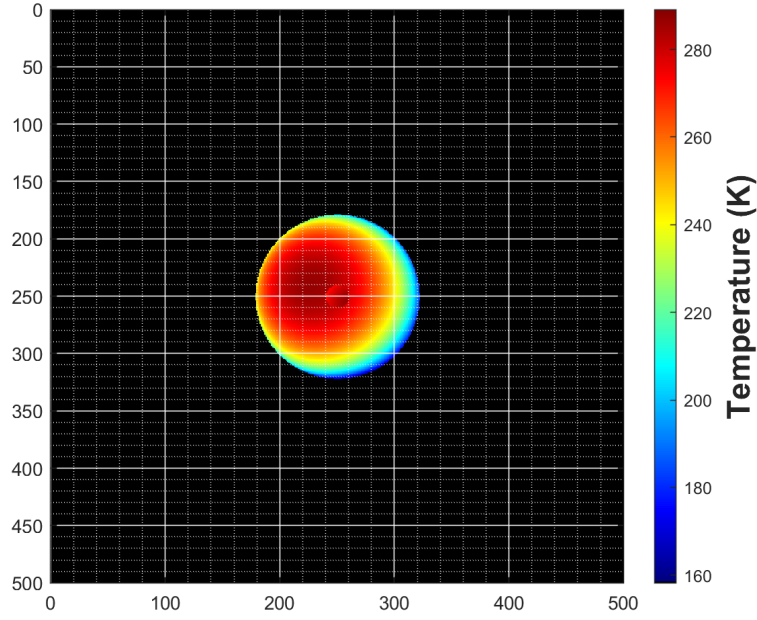


Figure 2.6: Rasterized Rendering Example Image (Image resolution is 500×500 pixels with a camera FOV of 5°)

This process is applied to each facet center to produce an array of projected pixels and associated temperatures. The temperatures of a given pixel is then interpolated by using a scattered interpolation scheme [9] based on the temperatures of nearby facet center pixels. An example of such an image is given in Fig. 2.6. Unlike with MATLAB's rendering, arbitrary resolutions and field-of-views can be simulated using the outlined process. This rendering scheme is used to generate simulated images for use in testing the algorithm presented in Chapter 6.

Chapter 3

Spin Pole Estimation

3.1 Introduction and Background

Pole estimates of small bodies are necessary to perform successful spacecraft operations in proximity of these bodies. The pole axis ${}_{\mathcal{I}}\hat{\mathbf{p}}$ defines the body frame (${}_{\mathcal{B}}\hat{\mathbf{z}}$ from Fig. 2.1) which is necessary to characterize the small body shape in addition to being useful for autonomous navigation relative to the surface (Terrain Relative Navigation, TRN).

As ground based sources of the pole axis produces inaccurate estimates and stereo photogrammetry (SPC) is computationally intractable and non-autonomous, there is a need for a more autonomous, computationally efficient algorithm. In this chapter, a method of extracting the pole from a set of infrared images of a small body is presented. In a set of infrared images, the geometry between the camera and the pole axis can be observed. This information is leveraged to produce an estimate of the pole axis. Similar algorithm to the one presented here has been proposed [97] which uses features extracted from a set of optical images. The algorithm presented doesn't require feature tracking and is more robust to illumination geometry as it uses infrared images.

3.1.1 Assumptions

The following are a list of assumptions made in the thermal modeling and pole axis estimation algorithms presented in this paper.

- (1) The camera attitude relative to the inertial frame is known and its boresight is pointed at the center of the asteroid.

- (2) The spin period of the asteroid is known. Light-curve observations made from the ground result in enough accuracy and therefore it is assumed to be the truth [34, 132].
- (3) The asteroid is assumed to be a principal axis rotator (at least over the spin period).
- (4) The approach distance is far enough from the asteroid such that its gravity does not influence the spacecraft motion.
- (5) Over the observational period, the apparent size of the asteroid (in the camera field of view) does not change significantly. The spacecraft is effectively hovering relative to the asteroid and the relative velocity along the boresight direction is negligible.

3.2 Outline

The rest of this chapter is outlined as follows. Section 3.3 presents the pole axis estimation method which includes a description of the geometry (Section 3.3.1), the image processing (Section 3.3.2), and the spin direction estimation method (Section 3.4). Section 3.6 presents the pole estimation algorithm results on simulated images of the asteroids Bennu (Section 3.6.1) and Itokawa (Section 3.6.2) and real images of asteroid Ryugu from JAXA’s Hayabusa2 mission (Section 3.6.3). Section 3.7 presents a discussion and evaluation of the pole estimation algorithm’s sensitivity to various parameters. Finally, Section 3.8 gives a concluding summary of the method and the results.

3.3 Spin Pole Axis Estimation

In this section, the pole axis and spin direction estimation methods are presented. The pole axis is defined as the line about which the body rotates. The spin direction is defined as the vector orthogonal to the pole axis which follows the motion of the body. Section 3.3.1 establishes the relationship between the latitude lines and the pole axis. Section 3.3.2 illustrates the image processing steps to obtain the latitude lines from a set of infrared images. Section 3.4 shows a

method to estimate the spin direction of the asteroid. Finally, Section 3.5 shows the algorithm pseudo-code.

3.3.1 Obtaining a Pole Estimate from Latitude Lines

Imagine the path of a point on the surface of an asteroid spinning about its pole axis. As the asteroid spins, this point traces out a circle of constant latitude (a *latitude line*). Such points could be rocks or other topographical features on the surface of the asteroid. Due to differences in thermal inertia or topology, gradients between latitudes appear when a set of infrared images are stacked. These gradients appear as long as there is a sufficient difference in temperature between parts of the surface. The appearance of these streaks in the images is dependent on the geometry between the pole axis and the camera. The proposed method utilizes this geometry to estimate the pole.

Consider one such latitude line of radius R on an asteroid as shown in Fig. 3.1a along with the pole axis $\mathcal{I}\hat{\mathbf{p}}$. An observer with a camera is at a fixed distance at $\mathcal{I}\mathbf{r}$ (Fig. 2.1) from the asteroid. Extruding the circular latitude line in the $\hat{\mathbf{r}}_{cam}$ ($\hat{\mathbf{r}}_{cam} = \frac{\mathcal{I}\mathbf{r}}{\|\mathcal{I}\mathbf{r}\|}$) direction forms an *oblique cylinder* (shown in Fig. 3.1b). The appearance of the latitude line from a camera is determined using the intersection between this oblique cylinder and the camera plane (shown in Fig. 3.1c). When the extrusion direction $\hat{\mathbf{r}}_{cam}$ is parallel to the pole axis $\mathcal{I}\hat{\mathbf{p}}$ and therefore perpendicular to the latitude line, the intersection will be the original latitude line circle of radius R . Otherwise, the intersection will be an ellipse. The shape of this projected ellipse can be parameterized by two semi axes: semi-major axis a and semi-minor axis b (ellipse is the black dotted line in Fig. 3.1c). The geometry between R and a is illustrated in Fig. 3.1d in which the projected cylinder is viewed along the pole axis $\mathcal{I}\hat{\mathbf{p}}$. This geometry shows that, regardless of the observer (camera) latitude ψ_{cam} , the semi-major axis a equals the latitude line radius R . The geometry between R and b is shown in Fig. 3.1e in which the R and b form a right-angle triangle (with R as the hypotenuse). The angle between these two values δ is computed from the observer (camera) latitude ψ_{cam} (Eq. (3.2)). From the geometry established in Figs. 3.1d and 3.1e, the equations relating the projected ellipse's

semi axes a and b to the latitude line radius R can be derived:

$$a = R \text{ (semi-major axis)} \quad (3.1)$$

$$\delta = \frac{\pi}{2} - \psi_{cam} \text{ (shown in Fig. 3.1e)} \quad (3.2)$$

$$b = R \cos \delta \text{ (semi-minor axis)} \quad (3.3)$$

Using Eq. (3.1) and Eq. (3.3), the eccentricity e of the projected ellipse can be computed:

$$e = \sqrt{1 - \frac{b^2}{a^2}} = \sqrt{1 - \frac{R^2 \cos^2 \delta}{R^2}} = \sqrt{1 - \cos^2 \delta} = \sin \delta \quad (3.4)$$

$$\implies e = \cos \psi_{cam} \quad (3.5)$$

Eq. (3.5) shows that the eccentricity of the ellipse e is **not** a function of the latitude line radius R . Thus, only ψ_{cam} determines the eccentricity of **any** projected latitude line. Therefore, by estimating the eccentricity of an ellipse from a set of images, an estimate of the observer latitude ψ_{cam} can be obtained.

The shape of this ellipse, as parameterized by e , is used to compute the observer latitude ψ_{cam} which is one of the two angles required to define the pole axis ${}_I\hat{\mathbf{p}}$ in $\mathbb{R}^3$. This eccentricity represents the component of ${}_I\hat{\mathbf{p}}$ projected along the camera boresight axis ${}_C\hat{\mathbf{z}}$. As ${}_C\hat{\mathbf{z}}$ is orthogonal to the camera plane, the other component of ${}_I\hat{\mathbf{p}}$ will lie in this plane. The projected ellipse in the camera plane is shown in Fig. 3.1f. Figs. 3.1d and 3.1e show that the projection of the pole axis onto the camera plane must lie along the minor axis $\hat{\mathbf{b}}$ of the projected ellipse. The angle ϕ between this vector $\hat{\mathbf{b}}$ and the camera x-axis ${}_C\hat{\mathbf{x}}$, shown in Fig. 3.1f, represents the component of the pole axis ${}_I\hat{\mathbf{p}}$ projected onto the camera plane. Therefore the two angles ψ_{cam} (or equivalently

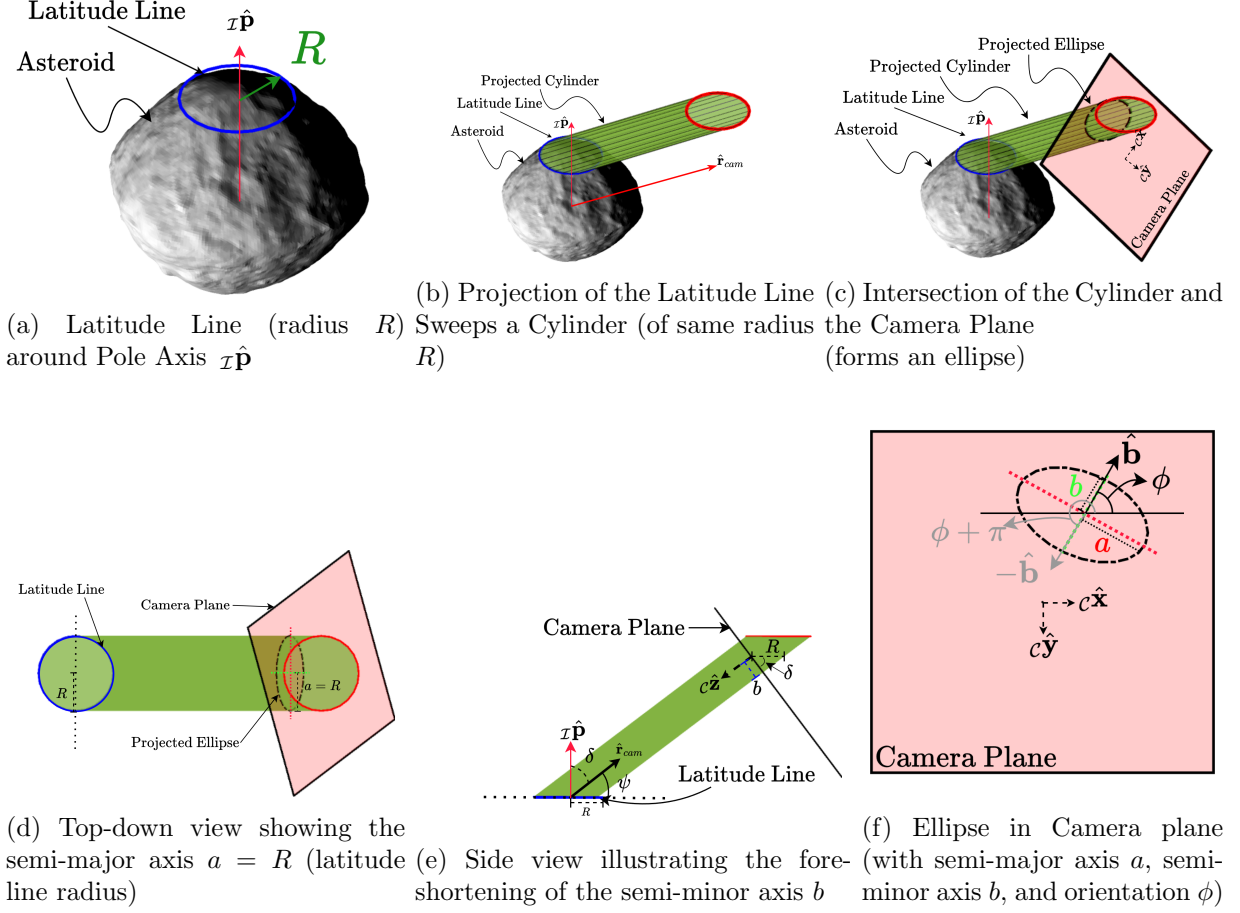


Figure 3.1: Visualization of the Latitude Line Geometry

e) and ϕ are used to define the pole axis $z\hat{\mathbf{p}}$. The following equations show this definition:

$$c\hat{\mathbf{z}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (\text{boresight direction in camera frame}) \quad (3.6)$$

$$c\hat{\mathbf{b}} = \begin{bmatrix} \sin \phi \\ \cos \phi \\ 0 \end{bmatrix} \quad (\text{minor axis of the ellipse in the camera frame } C) \quad (3.7)$$

$$c\hat{\mathbf{p}} = \pm c\hat{\mathbf{z}} \cos \delta \pm c\hat{\mathbf{b}} \sin \delta = \pm c\hat{\mathbf{z}} \cos \left(\frac{\pi}{2} - \cos^{-1} e \right) \pm c\hat{\mathbf{b}} e \quad (3.8)$$

$$c\hat{\mathbf{p}} = \pm c\hat{\mathbf{z}} \sqrt{1 - e^2} \pm c\hat{\mathbf{b}} e \quad (\text{pole axis in the camera frame including the two ambiguities}) \quad (3.9)$$

$$z\hat{\mathbf{p}} = {}^I_C \mathbf{R} c\hat{\mathbf{p}} \quad (\text{pole axis in the inertial frame}) \quad (3.10)$$

The first component of the sum in Eq. (3.9) represents the projection of the pole along the boresight axis ${}_C\hat{\mathbf{z}}$. The second component represents the projection of the pole onto the camera x-y plane along the minor axis direction $\hat{\mathbf{b}}$. Note the two ambiguities as indicated by $\pm[\cdot]$ in Eq. (3.9). The first of these, $\pm {}_C\hat{\mathbf{z}}$, indicates the ambiguity of the pole axis in relation to the boresight: the *boresight ambiguity*. The boresight ambiguity represents the fact that projections of the pole axis ${}_I\hat{\mathbf{p}}$ along either of the two vectors $+{}_C\hat{\mathbf{z}}$ and $-{}_C\hat{\mathbf{z}}$ are at the same angle ψ_{cam} and the same eccentricity e (using Eq. (3.5)). Thus, an eccentricity estimate e of the projected ellipse does not indicate whether the projection of the pole axis ${}_I\hat{\mathbf{p}}$ lies along the $+{}_C\hat{\mathbf{z}}$ or the $-{}_C\hat{\mathbf{z}}$ directions of the boresight axis. The boresight ambiguity is related to the *spin direction* and can be resolved by estimating this direction. A method for estimating the spin direction is given in Section 3.4.

The second of the ambiguities is in relation to the minor axis $\hat{\mathbf{b}}$: the *in-plane ambiguity*. This ambiguity represents the fact that both $+$ and $-$ directions along $\hat{\mathbf{b}}$ are at the *same* angle of ϕ from the x-axis of the camera frame ${}_C\hat{\mathbf{x}}$. Thus, estimating the orientation of the projected ellipse ϕ does not indicate whether the minor axis is at a counter-clockwise angle of ϕ or $\phi + \pi$ from ${}_C\hat{\mathbf{x}}$. Fig. 3.1f shows an illustration of the in-plane ambiguity in which both directions $\pm \hat{\mathbf{b}}$ are shown. A method for resolving this ambiguity is given in Section 3.3.2.

3.3.2 Obtaining Latitude Lines from Images

The relationship between latitude lines and the pole is established in the previous section. In addition, the equations relating the projection of these lines onto the camera plane to the pole axis are derived. Here, a robust process of obtaining these projected latitude lines (ellipses) is presented.

3.3.2.1 Image Processing Steps to Obtain Latitude Lines From Images

Consider a sequence of infrared images taken over one rotational period of the asteroid (Fig. 3.2a). These images are generated by applying the thermal model outlined in Section 2.2.1

on a shape model of the asteroid Bennu.¹ Stacking and then normalizing these images reveals a structure that follows the projected ellipses of the latitude lines (Fig. 3.2b). An investigation of the required image quantity and frequency for this method is presented in Section 3.7.2.

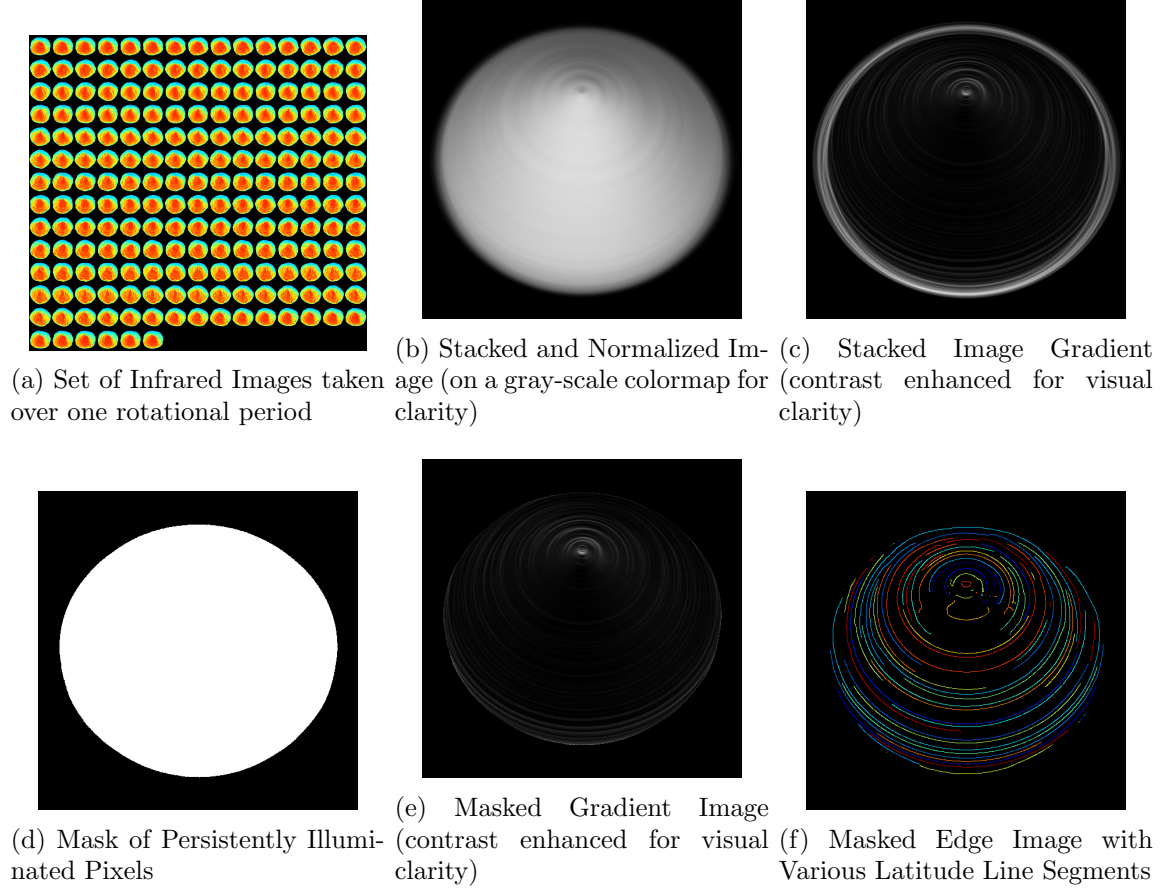


Figure 3.2: Procedure to Extract Latitude Lines from IR Images

In order to extract these ellipses, an image gradient of the stacked image is applied (using a Sobel operator) which identifies these latitude lines (Fig. 3.2c). Fig. 3.2c also shows a significant gradient around the silhouette of the asteroid. This occurs due to the large gradient between the asteroid and the background of space. Thus, the gradients within the asteroid silhouette appear weaker by comparison. This leads to poor performance of the extracted edges (using the canny algorithm). An image mask (Fig. 3.2d) representing the persistently illuminated image pixels is

¹ 75 cm shape model can be found [here](#)

used on the gradient image to remove this *silhouette gradient*. The processed gradient image is shown in Fig. 3.2e.

In the processed (masked) gradient image (Fig. 3.2e), the latitude lines appear in the form of several bands of gradients which follow the projected latitude line ellipses. A Canny edge operator [20] is applied to this masked gradient in order to segment these bands into individual streaks. It is observed that the silhouette gradient persists even after the application of the image mask. As a result of this, the extracted edges will still contain the undesirable silhouette edge. Reapplying the image mask to the edge image proves to be an effective mitigation of this issue. Fig. 3.2f shows this masked edge image in which each streak is assigned a different color. Each streak follows a projected ellipse of a latitude line. While the vast majority of streaks in this edge image conform to the expected ellipse shape, there remain a few streaks that do not. These stem from gradients within the stacked image that are not from the latitude lines. A few strategies are employed to improve the pole estimate and are discussed later in this section.

3.3.2.2 Fitting Ellipses to Latitude Lines

As stated in Section 3.3.1, an ellipse fit of each streak is necessary in order to estimate the pole axis. Many ellipse fitting algorithms are found in the literature such as direct methods [36, 101, 120] and iterative methods [4, 71, 77]. Consensus based methods [24, 38] (*e.g.* using random-sample-consensus RANSAC) can utilize either or both of these methods to improve the fit. Fitzgibbon et. al [36] present a direct ellipse fitting method based on a least-squares formulation which is shown to be highly computationally efficient (by more than an order of magnitude) [38] compared to the iterative and consensus based methods. Although the iterative and consensus based methods can improve performance (and employ the direct method in their algorithms' design), the direct method is employed in this work due to this computational efficiency.

Each ellipse fit is characterized in the camera plane by five parameters: the ellipse center (x and y coordinates), orientation of the minor axis ϕ , eccentricity e , and the semi-major axis a . The orientation ϕ and eccentricity e estimates of the ellipse(s) can thus be used to obtain an estimate

of the pole (using Eqs. (3.9) and (3.10)). Three mitigation strategies are employed to improve the robustness of the pole estimation algorithm.

3.3.2.3 Improving the Robustness of the Ellipse Fits

The identification of the streaks is affected by the image resolution: higher resolution images (with smaller pixels) and/or sub-pixel edge detection algorithms [44, 51, 75, 106, 133, 141] will allow for more accurate streak identification, which naturally leads to more data which is more easily fit by the elliptical arcs. Due to this phenomenon, smaller streaks (streaks spanning fewer pixels) contain less curvature information which affects the resultant ellipse fit. By contrast, larger streaks are more likely to overcome this issue and therefore result in a more precise ellipse fit. In order to improve the pole estimate, a weight is assigned to each ellipse fit based on its streak length (line 10 of Algorithm 2). These weights are used to compute a *weighted mean* eccentricity and orientation estimates which results in an improved pole estimate .

An ellipse fit can be obtained to a streak as long as it consists of five points (pixels). With such small number of points, the ellipse fit will be inaccurate. Generally, the fewer the points, the less accurate the ellipse fit. To mitigate this issue, a minimum streak length (`minLength` in line 6 of Algorithm 2) is imposed which excludes streaks that are too small for a reliable ellipse fit. These streaks correspond either to a very small span of a given latitude line or to other spurious gradients in the edge image. Thus, streaks containing less points than `minLength` are excluded from the rest of the pole estimation algorithm.

The third strategy of improving the robustness of the ellipse fits is based on the observation that two streaks appear to follow the same ellipse (corresponding to the same latitude line). Such streaks appear in the edge images (Fig. 3.2f) as symmetric complements of each other about the minor axis of the ellipse. The ellipse fits of these individual streaks are less accurate compared to the ellipse fit on a combination of the two complementary streaks. Fig. 3.3 illustrates the streak matching procedure. The complementary streaks are identified by first computing the average minor axis direction $\hat{\mathbf{b}}_{avg}$ across all streaks (vertical blue line in Fig. 3.3). For a given streak that

is entirely on one side of this axis (red streak in Fig. 3.3), the distance from the median of this streak to the average minor axis $\hat{\mathbf{b}}_{avg}$ is computed (denoted as D in Fig. 3.3). A streak at this distance (within a threshold of `matchThresh`) on the opposite side of the average minor axis is identified as the complementary streak (green streak in Fig. 3.3). An ellipse fit is then applied to this combined streak. This procedure is labeled `matchStreaks()` on line 9 of Algorithm 2.

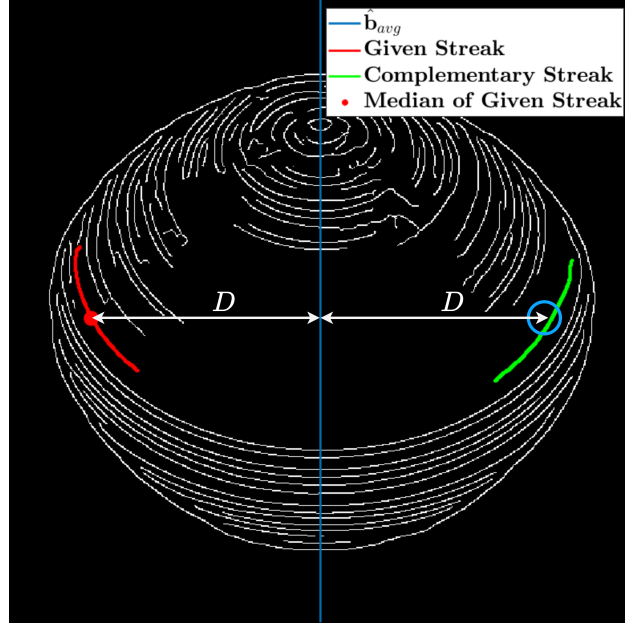


Figure 3.3: Procedure to combine complementary streaks. Note that the blue circle denotes the match radius threshold (`matchThresh`)

As the pole estimate is based on the eccentricity e and orientation ϕ of the ellipses (Eq. (3.9)), adding robustness (with these three strategies) to the ellipse fits will lead to a better pole estimate.

3.3.2.4 Resolving the In-Plane Ambiguity

The *in-plane ambiguity* was illustrated in Section 3.3.1. Here, this ambiguity is mitigated by estimating the curvature direction of streaks. Given a streak $\mathbf{S}_i$ and its ellipse fit, the curvature direction is defined as the vector from the ellipse center to the streak points (blue vectors in Fig. 3.4). This method is applied to each streak point across all streaks. The mean curvature direction is computed using all of the streaks (red vector in Fig. 3.4). The in-plane ambiguity is thus resolved

by choosing the minor-axis direction ($+\hat{\mathbf{b}}$ or $-\hat{\mathbf{b}}$) with the smaller angle with respect to the mean direction. In the example shown in Fig. 3.4, $-\hat{\mathbf{b}}$ is chosen.

When the streak $\mathbf{S}_i$ spans the entire ellipse, the mean curvature method is unsuitable to resolve the in-plane ambiguity. This situation occurs at polar latitudes when the eccentricity is 0 (Eq. (3.5)) and the minor-axis is undefined. Therefore, the need to resolve the in-plane ambiguity is eliminated.

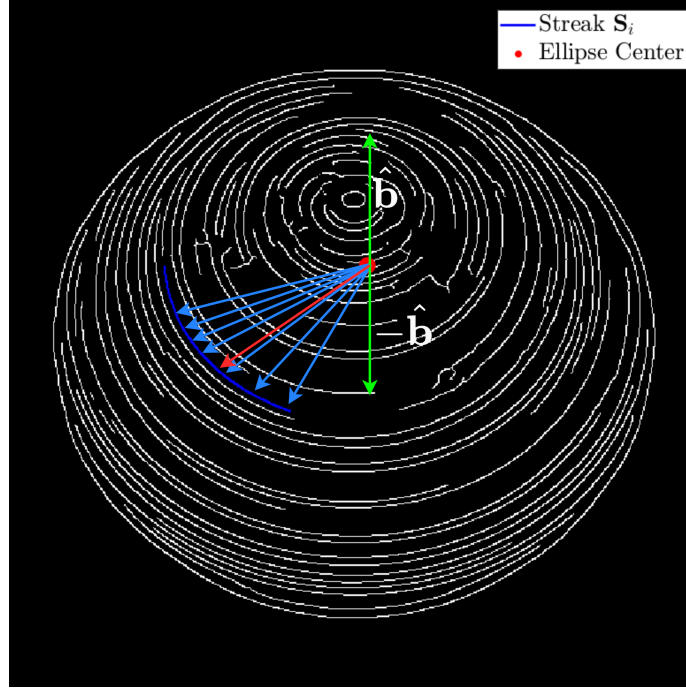


Figure 3.4: Illustration of the In-Plane Ambiguity Resolution

3.4 Spin Direction Estimation

In this section, a robust method to estimate the spin direction is presented. By estimating this spin direction, the *boresight ambiguity* is resolved.

3.4.1 Appearance of the Spin Direction in the Image

Consider the picture shown in Fig. 3.5a where the streaks (projection of the latitude lines) are shown in addition to the ellipse fit of a particular streak (red ellipse). The spin direction can

be determined by observing the direction along the ellipse in which a point on this streak moves. There are two possibilities for this *direction of motion* $\mathbf{d}$ within the image plane: $\mathbf{d}_1$ and $\mathbf{d}_2$ as shown in Fig. 3.5b. The center of this ellipse is the pivot-point around which the $\mathbf{p}$ rotates. Since the vectors $\mathbf{p}$, $\mathbf{d}_1$, and $\mathbf{d}_2$ are in the camera plane, the cross product between $\mathbf{p}$ and either of the direction vectors will lie along the boresight direction: $\mathbf{p} \times \mathbf{d}_1$ lies along the $+\mathcal{C}\hat{\mathbf{z}}$ and $\mathbf{p} \times \mathbf{d}_2$ lies along the $-\mathcal{C}\hat{\mathbf{z}}$ direction. By utilizing this relationship, the spin direction can be estimated from a pair of images which is then used to resolve the boresight ambiguity.

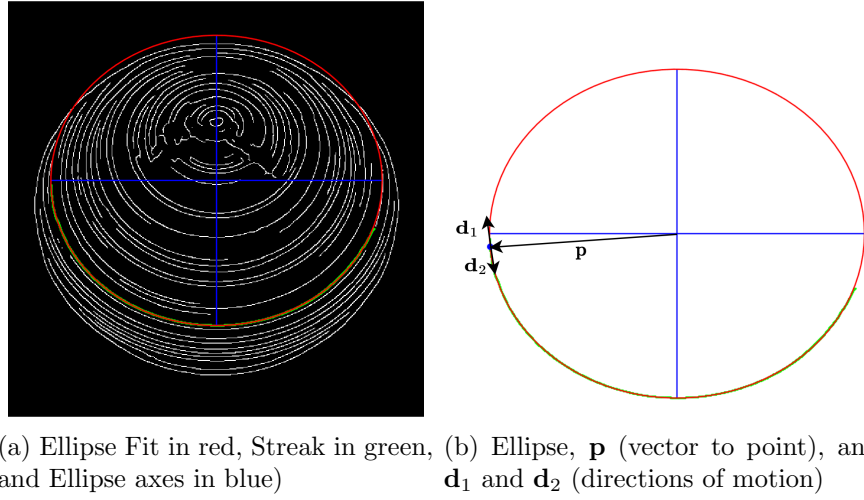


Figure 3.5: Spin Direction Illustration (Fig. 3.5a shows an edge image extracted from stacking images)

3.4.2 Obtaining the Direction of Motion from Two Images

In this section, a process to extract the direction of motion and the pivot vector from two images (out of the stack of IR images) is presented. The *direction of motion* ($\mathbf{d}$) can be extracted from a pair of images using standard feature descriptors (such as SIFT, SURF, KAZE etc.). However, depending on the resolution of the image, these may not be reliable. Therefore, a pseudo feature detection and tracking approach is presented in this section that can robustly detect the motion direction between a pair of images. The method is based on the assumption that between a consecutive pair of images, similar *edge features* appear at different locations due to the motion

of the body. An *edge feature* is defined as a connected sequence of pixels as detected by an edge detection algorithm (the Canny algorithm is employed in this work). Thus, the *direction of motion* $\mathbf{d}$ is encoded in the movement of these edges. The steps of this direction of motion estimation method are summarized as follows (and illustrated in Fig. 3.6).

- (1) **Obtain edges from images:** Apply the Canny edge operator to each image in a pair of consecutive IR images to extract edge features (result shown in Fig. 3.6a).
- (2) **Associate edge features between two images:** For each edge feature in the first image, compute the euclidean distance to the edge features in the second image (between the centroids of the features). The edge feature in the second image with the shortest euclidean distance is associated with the edge feature in the first image. Therefore, each edge feature in the first image will likely be associated with an edge feature in the second image that is closest to it. One such pair of edge features is highlighted by the red box in Fig. 3.5
- (3) **Estimating direction of motion $\mathbf{d}$:** The direction of motion is the vector between the median locations of the two associated features which is labeled $\mathbf{d}$ in Fig. 3.6c.
- (4) **Estimating the vector $\mathbf{p}$:** For a given edge feature in the first image which has a corresponding edge feature in the second image, compute the euclidean distance between the feature centroids and all streak points. The vector $\mathbf{p}$ is then defined as the intersection point between the streak and the edge feature from the first image. In Fig. 3.6b, this intersecting streak is shown in addition to the ellipse fit of the streak which is indicated by its center. These ellipses are obtained from the stacked images as discussed in Section 3.3.2.

From each edge feature pair, the vectors $\mathbf{d}$ and $\mathbf{p}$ are computed. As described in Section 3.4.1, taking the cross product of these two vectors gives the spin direction and resolves the boresight ambiguity. As the feature association may result in incorrect associations, the direction of motion estimate is not likely to be the same from each edge feature pair. In order to obtain a robust

spin direction, this process is applied to all edge pairs in all pairs of consecutive images across the available image set (nominally over one spin period of the asteroid). The spin direction is thus the direction that occurs the most (i.e. the mode).

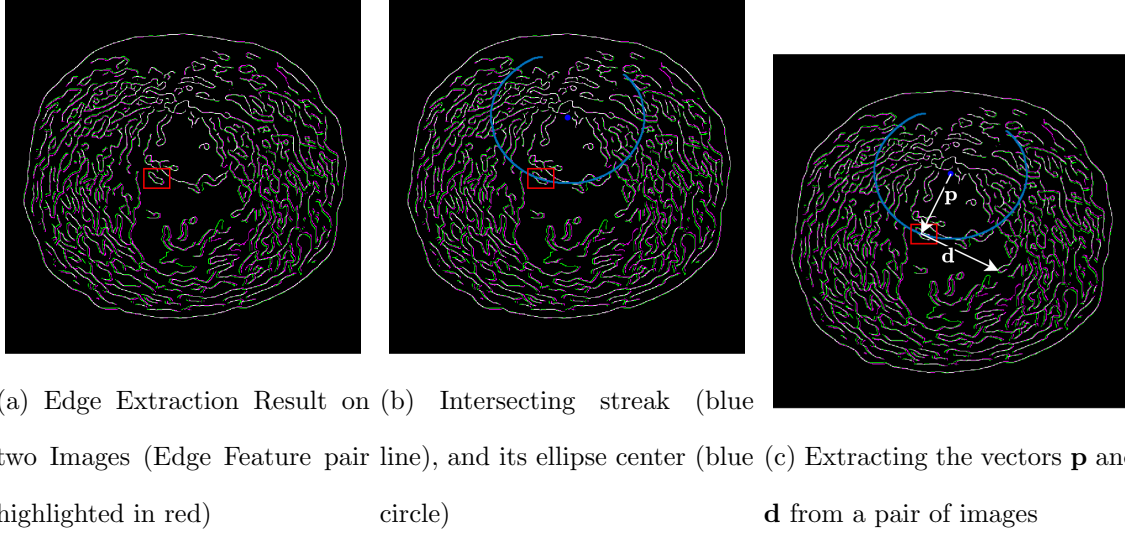


Figure 3.6: Identifying Spin Direction (previous edge in green, current edge in pink)

This method of obtaining the spin direction and thus resolving the ambiguity along the boresight axis works as long as the vectors $\mathbf{p}$ and $\mathbf{d}$ are not parallel. They are parallel at an equatorial observer latitude of ($\psi_{cam} = 0^\circ$). In this case, the ambiguity doesn't exist since the decomposition of the pole axis onto the boresight axis is zero. As the observer latitudes get close to the equator, the $\mathbf{p}$ and $\mathbf{d}$ vectors become more parallel which renders the disambiguation method to be less reliable. Therefore, an alternate method must be used for such cases to guarantee robust performance. At observer latitudes close to the equator, the cross product between the $\mathbf{d}$ vector and the pole axis projection onto the minor-axis (with the in-plane ambiguity resolved) provides information that can be used to resolve the boresight ambiguity. This cross product will once again lie either along the $+\mathcal{C}\hat{\mathbf{z}}$ or $-\mathcal{C}\hat{\mathbf{z}}$ direction of the boresight axis. Thus, this provides an alternate method of resolving the boresight axis ambiguity.

If the estimated observer latitude (Eq. (3.5)) is found to be close to the equator, then this

alternate boresight ambiguity resolution is applied. It is observed that either method is effective for mid-latitudes ($\sim 20^\circ$ to 60°) indicating that the threshold latitude to decide between the two methods can be fairly coarse. In practice, setting this threshold to approximately 20° produced robust ambiguity resolution performance.

3.5 Spin Pole Estimation Algorithm

In this section, the pseudo-code of the spin pole estimation algorithm is given (Algorithm 2) which summarizes the key steps of the algorithm discussed in Sections 3.3 and 3.4. Lines 1–5 of the algorithm show the image processing steps to isolate the streaks. Lines 6–9 produce the ellipse fits. Lines 10–12 compute the weighted orientation and eccentricity means. Lines 13–18 resolve the *in-plane ambiguity* using the method from Section 3.3.2. Line 19 estimates the spin direction using the method from Section 3.4. Lines 20–24 uses the spin direction to resolve the *boresight ambiguity* and compute the pole estimate in the camera frame ${}_C\hat{\mathbf{p}}$. Line 25 converts the pole axis estimate ${}_C\hat{\mathbf{p}}$ into the inertial frame.

Algorithm 2 Pole Axis Estimation Algorithm**Given:** I_i , matchThresh, minLength, cannySig, N **Result:** ${}^{\mathcal{I}}\hat{\mathbf{p}}$

```

1:  $I_{mean} = \frac{1}{N} \sum_{i=1}^N I_i$  ▷ Pole estimate in the inertial frame
2:  $I_{mask} = \prod_{i=1}^N (I_i > 0)$  ▷ Stack and normalize  $N$  IR images
3:  $I_{grad} = \text{imgradient}(I_{mean}) \odot I_{mask}$  ▷ Image mask of persistently lit pixels
4:  $BW = \text{cannyEdge}(I_{grad}, \text{cannySig}) \odot I_{mask}$  ▷ Gradient of the image gradient (mask applied element-wise)
5:  $S = \text{segmentStreaks}(BW)$  ▷ Extract edges using the Canny operator (apply edge mask)
6:  $ell = \text{fitEllipse}(S, \text{minLength})$  ▷ Segment streaks from edge image
    $\text{minLength threshold}$  ▷ Use DLS algorithm [36] to fit ellipses to streaks that meet the
7:  $\phi_{nom} = \text{mean}(ell.\phi)$  ▷ Unweighted mean of the ellipse orientation
8:  ${}_c\hat{\mathbf{b}} = [\cos(\phi_{nom}), \sin(\phi_{nom})]^T$  ▷ Minor axis of ellipse in camera frame
9:  $ell = \text{matchStreaks}(S, {}_c\hat{\mathbf{b}}, \text{matchThresh}, \text{minLength})$  ▷ Match ellipses within threshold
10:  $w = \frac{\text{length}(S)}{\sum \text{length}(S)}$  ▷ Weighted by length of streak relative to others
11:  $\phi_{est} = \sum w \odot ell_i.\phi$  ▷ Weighted mean
12:  $e_{est} = \sum w \odot ell_i.e$  ▷ Weighted mean
13:  $p_v = ell.y - ell.y_0$  ▷ Streak points relative to the ellipse center (curvature direction)
14:  $p_{v,avg} = \text{mean}(p_v)$  ▷ Average curvature direction
15:  ${}_c\hat{\mathbf{b}} = [\cos(\phi_{est}), \sin(\phi_{est})]^T$  ▷ Initial minor axis estimate
16: if  $p_{v,avg} \cdot {}_c\hat{\mathbf{p}} < 0$  then ▷ Resolving the in-plane ambiguity
17:    ${}_c\hat{\mathbf{b}} = -{}_c\hat{\mathbf{b}}$  ▷ Updated minor axis estimate
18: end if
19:  $[\mathbf{p}, \mathbf{d}] = \text{estimateSpinDir}(I, ell, \text{cannySig})$  ▷ Estimate Spin Direction (Section 3.4)
20: if  $\text{sign}((\mathbf{p} \times \mathbf{d}) \cdot {}_c\hat{\mathbf{z}}) > 0$  then ▷ Resolve boresight ambiguity
21:    ${}_c\hat{\mathbf{p}} = -{}_c\hat{\mathbf{z}}\sqrt{1 - e_{est}^2} + {}_c\hat{\mathbf{b}}e_{est}$  ▷ Estimated Pole Axis in the camera frame (Eq. (3.9))
22: else
23:    ${}_c\hat{\mathbf{p}} = +{}_c\hat{\mathbf{z}}\sqrt{1 - e_{est}^2} + {}_c\hat{\mathbf{b}}e_{est}$  ▷ Estimated Pole Axis in the camera frame (Eq. (3.9))
24: end if
25:  ${}^{\mathcal{I}}\hat{\mathbf{p}} = \frac{\mathcal{I}}{c}\mathbf{R}{}_c\hat{\mathbf{p}}$  ▷ Estimated Pole Axis in the inertial frame (Eq. (3.10))

```

3.6 Results

In this section, the pole estimation and spin direction methods are applied to simulated infrared images of the asteroid Bennu (Section 3.6.1) and the asteroid Itokawa (Section 3.6.2) at various axial tilts and observer latitudes (γ and ψ_{cam} in Fig. 2.1). The axial tilt will determine the size of the region around the pole that is in permanent shadow or illuminated state which

will affect the images. The simulated images are generated using the thermal model outlined in Algorithm 1 for which the following warming and cooling rates are used: $r_w = 1/30$, $r_c = 1/100$. The simulated images have a high phase angle ϕ (see Eq. (2.1) and Fig. 2.1 for definition and illustration respectively) between 70° and 110° . The parameters of the camera used to generate simulated images are shown in Table 3.1. These images were generated using MATLAB’s OpenGL rendering engine. The spacecraft is assumed to be hovering relative to the body and the camera is pointing at the center of the body.

Table 3.1: Camera Parameters for the Simulated Bennu and Itokawa Images

Parameter Name	Parameter Value
Resolution	550×550 pixels
Field-of-View (FOV)	0.3°

In addition, the method is applied to real infrared imagery of asteroid Ryugu from JAXA’s Hayabusa2 mission [10, 93] (Section 3.6.3). The pole estimate errors are characterized as an angle between the true and the estimated pole axes (Eq. (3.11)):

$$\theta_{err} = \cos^{-1} (\mathcal{I}\hat{\mathbf{p}}^{est} \cdot \mathcal{I}\hat{\mathbf{p}}^{true}) \quad (3.11)$$

3.6.1 Pole Estimation Method Applied to Simulated Bennu Images

The thermal model described in Section 2.2.1 is applied to a 200,000 facet shape model of asteroid Bennu [16] to generate simulated infrared images over one rotational period of Bennu. The pole estimation method (Algorithm 2) is applied to these images. The number of images (N), spacing between images (ΔI) and other algorithm parameters used in this section are shown in Table 3.2. The sensitivity of the algorithm to these parameters is discussed in Section 3.7.2. As the rotational period of Bennu is 4.43 hrs [50], the baseline spacing of 1.8° which means that each pair of images are spaced 1.32 min apart.

The pole errors are shown in Fig. 3.7a for various axial tilt and observer latitude combinations. A histogram distribution of the errors is shown in Fig. 3.7b along with the mean and median error.

Table 3.2: Pole Estimation Algorithm Parameters for Simulated Bennu Images

Parameter Name	Parameter Value
matchThresh (pixels)	10
minLength (pixels)	50
cannySig	3
N	201
ΔI ($^\circ$)	1.8

The average error across all the cases is 2.1853° and approximately 95% of the cases have errors within the ground based uncertainty of 5° [91]. The average error for each axial tilt studied here is statistically similar ($\sim 2^\circ$) indicating that the algorithm may not be sensitive to this parameter. However, further analysis is warranted to confirm whether this insensitivity extends to other axial tilts. The errors are consistently larger at polar observer latitudes (around $\pm 90^\circ$). This is mostly due to the cosine relationship between the eccentricity and the observer latitude (Eq. (3.5)). This relationship makes it highly sensitive to eccentricity errors at latitudes close to $\pm 90^\circ$. Therefore, even a small eccentricity error maps to a significant error in the pole axis.

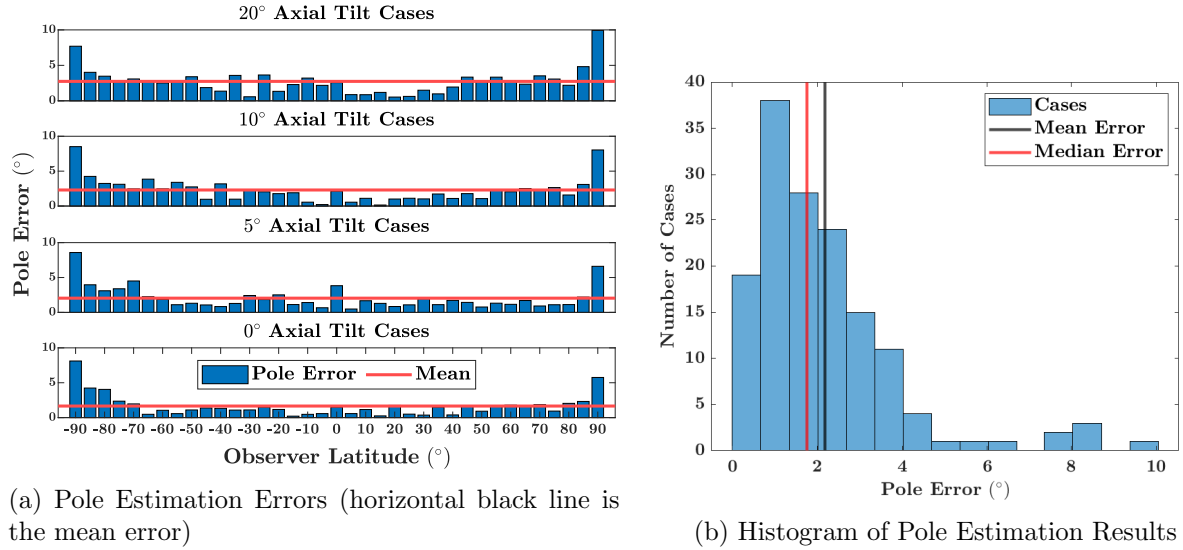


Figure 3.7: Pole Estimate Results on Simulated Infrared Images of Bennu

The ellipse fitting algorithm contributes significantly to the overall performance shown in Fig. 3.7. The eccentricity e and orientation ϕ of the fitted ellipse are the two parameters that are

used in estimating the the pole axis. Therefore, any errors in these parameters translate into errors in the resultant pole estimate. First consider Fig. 3.8 where the eccentricity estimates (Fig. 3.8a) and eccentricity errors (Fig. 3.8b) are shown. Each estimate corresponds to an individual streak for which an ellipse fit is obtained. The eccentricity estimates approximately follow a sinusoidal

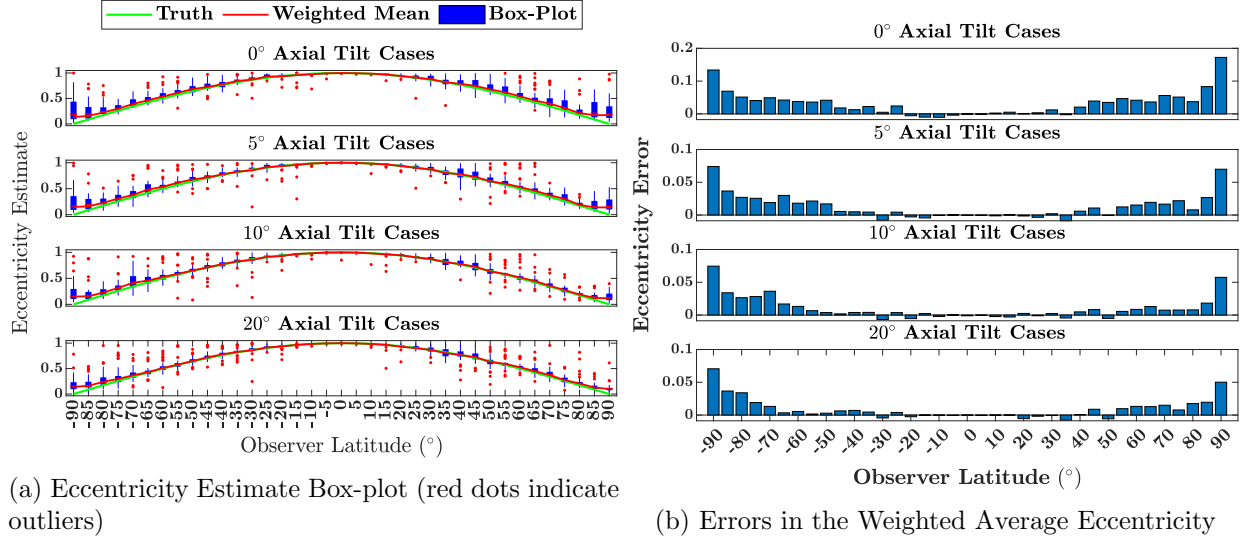


Figure 3.8: Eccentricity Estimate Performance

pattern as expected by Eq. (3.5). The box-plot in Fig. 3.8a shows larger variance in the estimates at polar observer latitude cases and the eccentricity error in these cases is larger in comparison. The ellipse fits at these latitudes should be nearly circular (low eccentricity). The larger errors in these polar cases indicates that the ellipse fitting algorithm is deficient for low eccentricity ellipses. The outliers in these box-plots are determined by the distance of a given estimate from the (unweighted) median and correspond to the ellipse fits of the smaller streaks. In contrast, the equatorial cases show smaller variance in the eccentricity estimates and the resultant error is smaller (as seen in the bar plots in Fig. 3.8b). These eccentricity errors correlate well with the pole estimate results from Fig. 3.7a. This implies that the eccentricity performance dictates the pole estimate performance. The orientation performance (shown in Fig. 3.9b) shows similar performance degradation at polar latitudes. However, at these polar latitudes, the orientation error does not contribute significantly to the pole estimate due to the relationship between the orientation and the pole axis (see Eq. (3.9)).

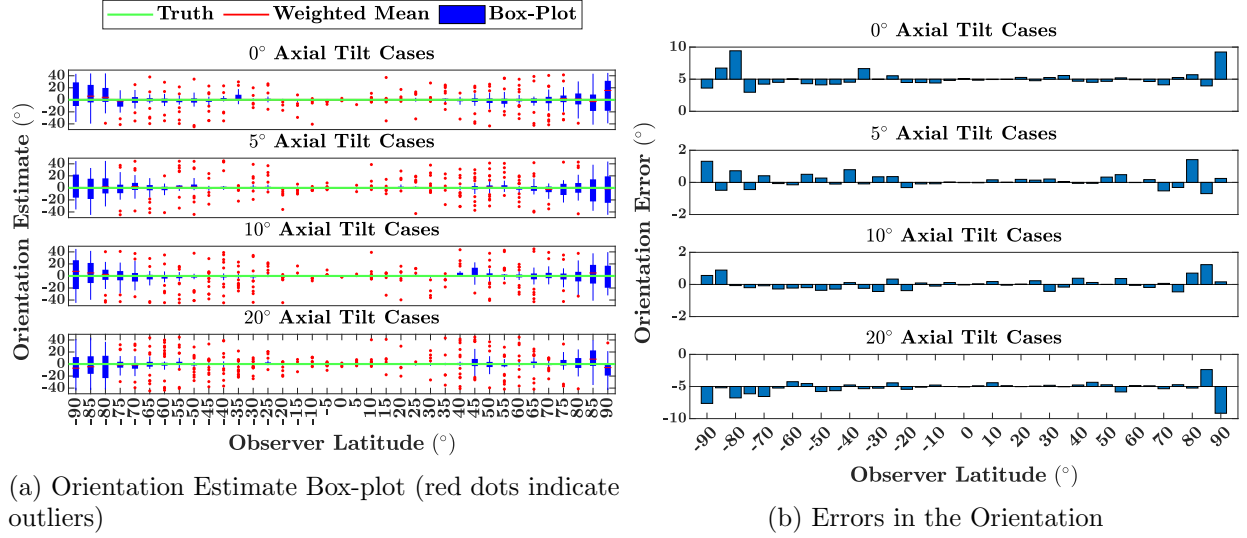


Figure 3.9: Orientation Estimate Performance

The true orientation ϕ is defined as the angle between the x-axis of the camera frame and the projection of the true pole axis into the camera plane. Thus, the orientation error is defined as the deviation from this true value.

3.6.2 Pole Estimation Method Applied to Simulated Itokawa Images

This section presents the results of the pole estimation algorithm on simulated IR images of the asteroid Itokawa. These images were obtained by using a 200,000 facet shape model of Itokawa [41]. Table 3.3 shows the list of algorithm parameters used. All of the parameters remain the same as the ones from Table 3.2 apart from the standard deviation used by the Canny edge algorithm (`cannySig`). The irregular shape causes the masking step to leave a comparably smaller edge image which results in fewer streaks if using the original standard deviation parameter.

At certain observer latitudes, the streaks outside masked region do not follow the latitude lines and therefore will result in a worse ellipse fit and pole estimate. Fig. 3.10 shows two edge images of Itokawa at observer latitudes of 80° and 30° in which the image masks are shown in pink, the masked streaks are shown in white, and the unused streaks are shown in green. In Fig. 3.10a, all streaks (both in white and green) follow the latitude lines. As these streaks outside of the masked

region are larger, including them in the estimation process would improve the pole estimate. In contrast, the green unused streaks in Fig. 3.10b (which are outside of masked region) do not follow the latitude lines. Therefore, their inclusion would worsen the pole estimate. To avoid this varying behavior (which is dependent on the observer latitude ψ_{cam} and the body shape) the green streaks are excluded from the pole estimation process (and therefore labeled unused in the Fig. 3.10). In order to extract more streaks from within the masked region (shown in pink in Fig. 3.10), the `cannySig` is lowered to 2.

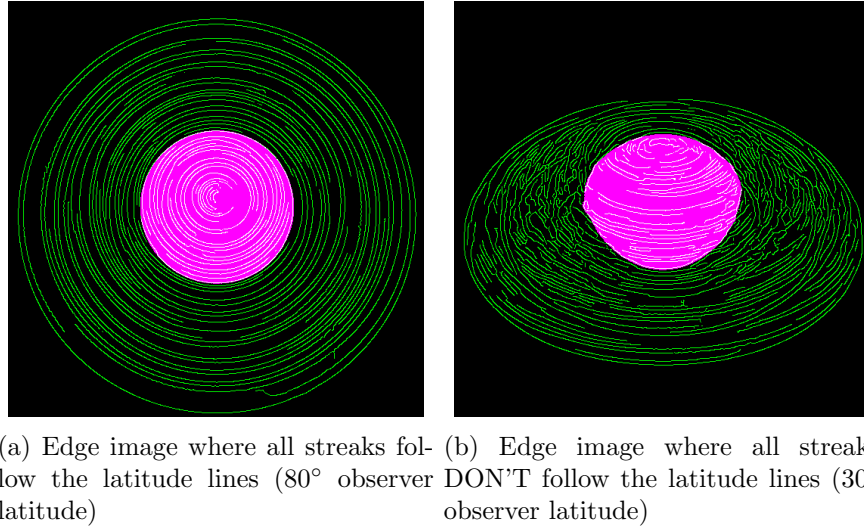


Figure 3.10: Itokawa Edge Images Comparison (pink region shows used streaks, unused streaks in green)

Table 3.3: Pole Estimation Algorithm Parameters for Simulated Itokawa Images

Parameter Name	Parameter Value
<code>matchThresh</code> (pixels)	10
<code>minLength</code> (pixels)	50
<code>cannySig</code>	2
N	201
ΔI ($^{\circ}$)	1.8

The pole estimation algorithm is applied to cases with various axial tilts and observer latitudes and the results are shown in Fig. 3.11. The mean error across all of the cases is 6.4496° and the

median error is 6.0887° as shown in Fig. 3.11b with approximately 91% of the cases having errors within the ground based uncertainty of 12° [57]. The worse performance is still observed (as it is with Bennu) at polar observer latitudes. This is a consequence of the masking step which results in fewer streaks. It is possible to improve these estimates by tuning the algorithm parameters (Table 3.3). Specifically, reducing the Canny standard deviation would admit more streaks. However, this would also allow for other, undesirable gradients to corrupt the data which could render the ellipse fits (and therefore the pole estimate) less accurate.

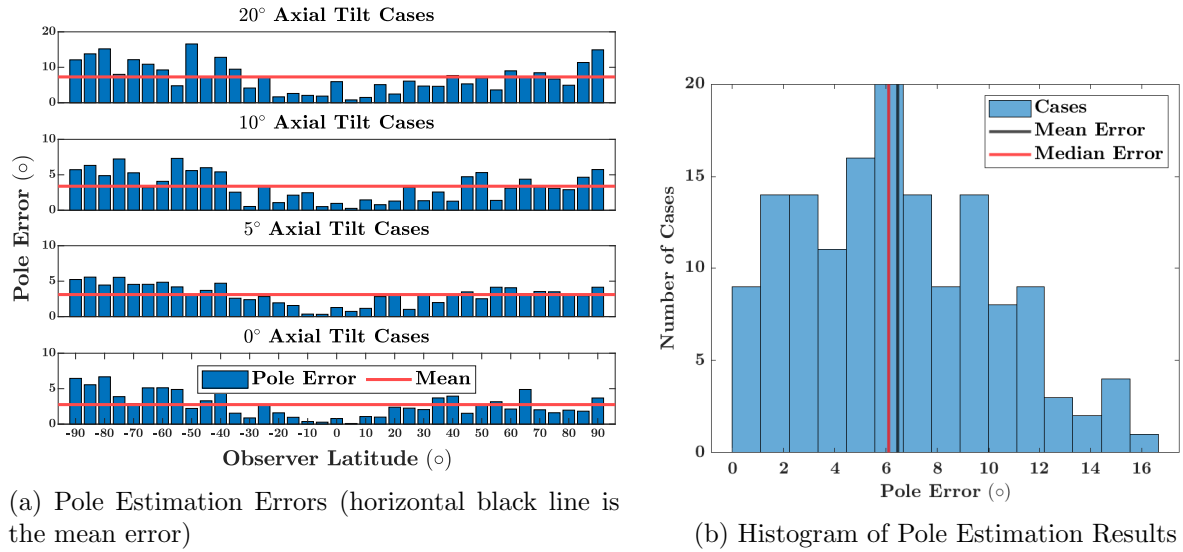


Figure 3.11: Pole Estimate Results on Simulated Infrared Images of Itokawa

As with the results on asteroid Bennu, the performance of the ellipse fitting algorithm contributes significantly to the overall pole error. To characterize this, the errors in estimating the eccentricity and orientation of the projected ellipses are shown in Figs. 3.12 and 3.13 respectively. The eccentricity estimates as a function of the observer latitude follows the expected sinusoidal pattern (following Eq. (3.5)). The performance at equatorial latitudes is similar to that seen with Bennu (Fig. 3.8a) but the performance at the polar latitudes is comparably worse. This explains the difference in pole estimate performance between Bennu (Fig. 3.7a) and Itokawa (Fig. 3.11a).

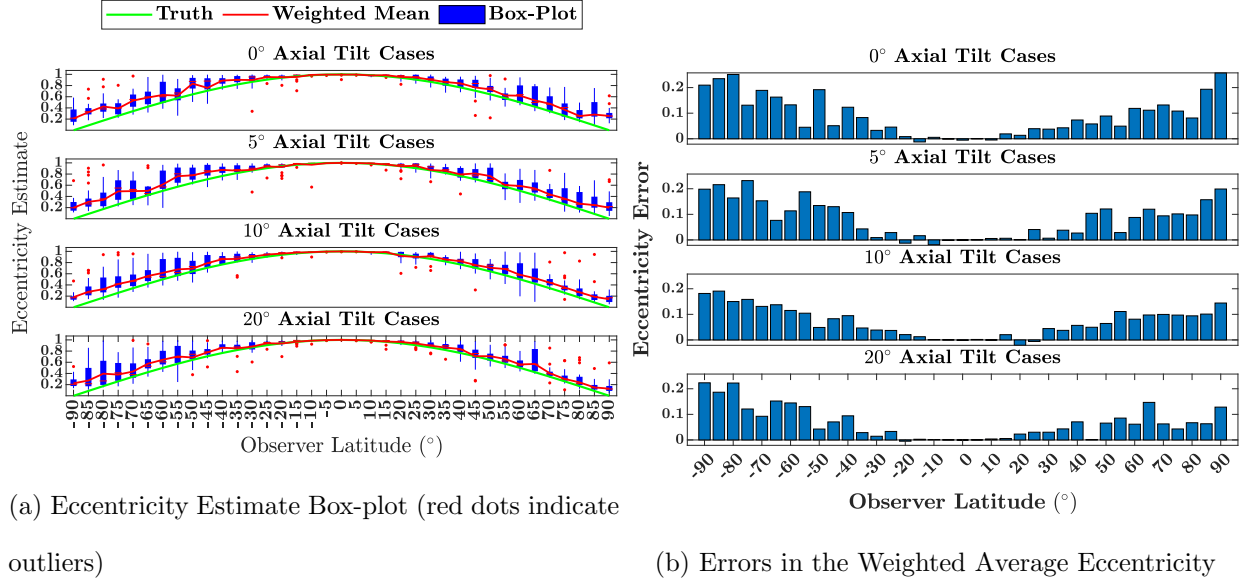


Figure 3.12: Itokawa Eccentricity Estimate Performance

The orientation estimates (Fig. 3.13a) are worse at polar latitudes (near $\pm 90^\circ$). This Itokawa performance is worse than the Bennu performance with the largest performance disparity occurring at polar observer latitudes. However, as observed with Bennu (Section 3.6.1), these orientation errors do not affect the resultant pole estimate.

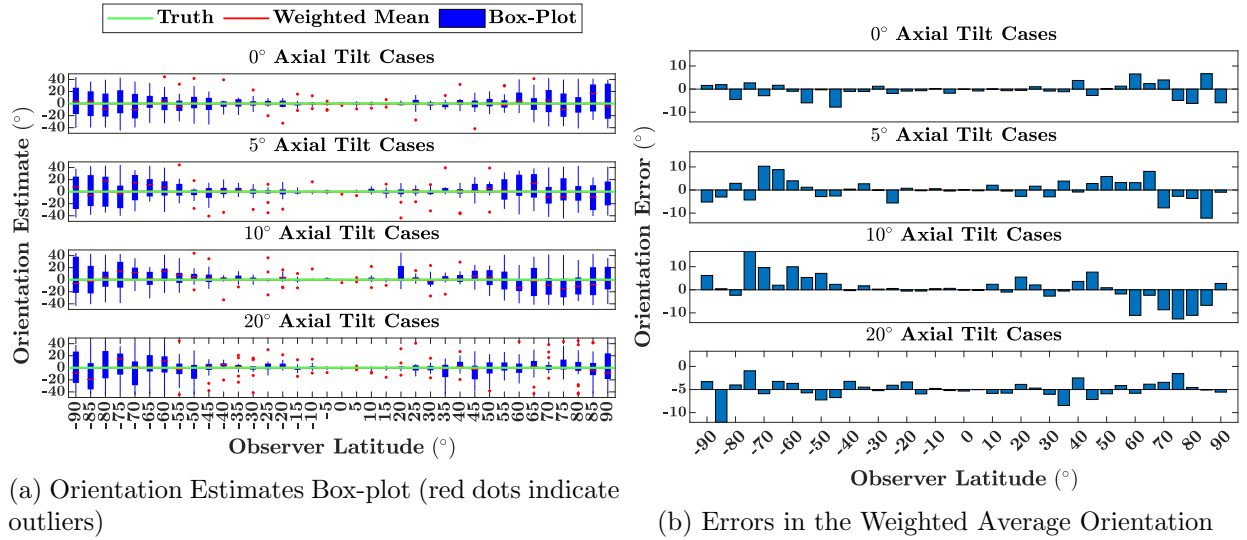


Figure 3.13: Itokawa Orientation Estimate Performance

3.6.3 Pole Estimation Method Applied to Hayabusa2 Images of Ryugu

The TIR camera [10] on the Hayabusa2 spacecraft took images [94] of the asteroid Ryugu while remaining in its home position approximately 20 km from the asteroid center. These images can be used to test the pole estimation algorithm on real infrared images from a mission. Fig. 3.14a shows a set of infrared images taken over one rotational period of Ryugu on October 20, 2018 [94]. In these images, the background appears blue as it is much colder than the asteroid. This contrast is used to remove the background from the infrared images. This is accomplished by applying a Canny operator over the image which detects the silhouette of the body. A mask generated from this silhouette is applied to the images to isolate the asteroid from the original set of images (see Fig. 3.14b). The images shown are cropped in to highlight the body.

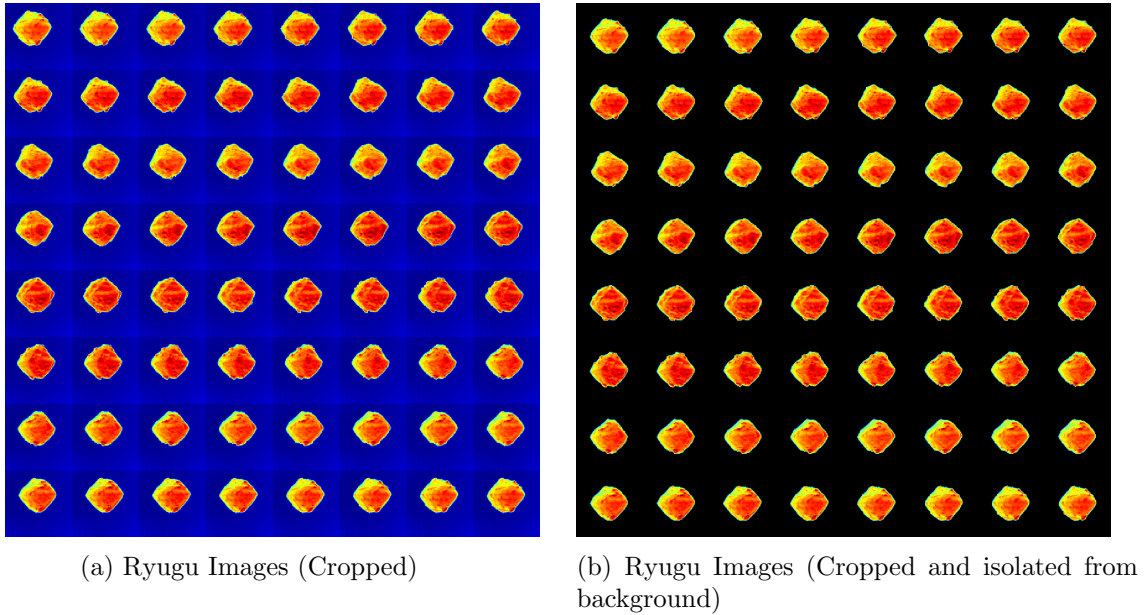


Figure 3.14: Hayabusa2 Infrared Images of Asteroid Ryugu

The algorithm parameters are summarized in Table 3.4. As the Ryugu image resolution is lower (simulated Bennu and Itokawa image resolutions are 550×550 , Ryugu image resolution is 248×328), the minimum streak length parameter (`minLength`) is set to 5 which is the minimum points required to define an ellipse.

Table 3.4: Pole Estimation Algorithm Parameters For Hayabusa2 Images of Ryugu

Parameter Name	Parameter Value
<code>matchThresh</code> (pixels)	5
<code>minLength</code> (pixels)	5
<code>cannySig</code>	0.5
N	64
ΔI ($^{\circ}$)	5.635°

The set of images shows that the relative position between Ryugu and the Hayabusa2 spacecraft is not constant. Therefore, to produce a useful stacked image, the images are aligned by their centers of brightness (CoBs). This step demonstrates the advantage of the IR images over optical images since the CoB from an IR image is much closer to the center of mass (CoM) of the body (especially at non-zero phase angles). This stacked image is shown in Fig. 3.15a. The procedure illustrated in Fig. 3.2 is applied to the stacked image and the resulting edge image is shown in Fig. 3.15e. It is interesting that the silhouette edge still persists despite the multiple application of the image mask indicating that there is deficiency in this heuristically derived step. Since none of the segments meet the minimum length threshold, they are not used for ellipse fitting. In addition, there are no streaks that satisfy the symmetry (about the minor axis) and therefore none of them are matched with each other.

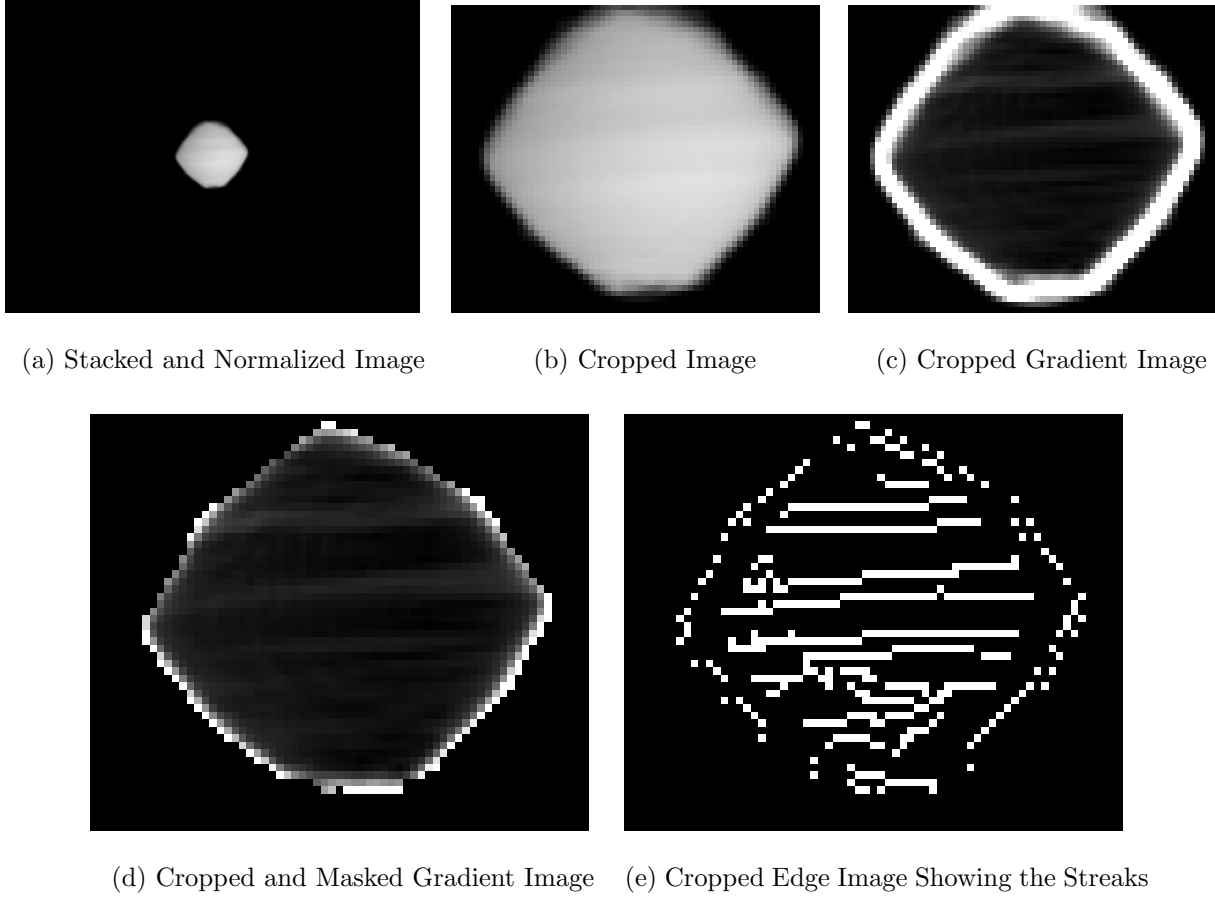


Figure 3.15: Obtaining Streaks from Hayabusa2 Images ((b)-(e) cropped to highlight asteroid)

The weighted eccentricity and orientation estimates of the ellipses along with the pole estimate are presented in Table 3.5. The true pole shown in the table is calculated using the attitude information from the Hayabusa2 SPICE [2, 3] Kernels². Using the SPICE attitude kernels, the true observer latitude ψ and the true minor axis orientation ϕ are computed. The true observer latitude ψ is then used to compute the true eccentricity e using Eq. (3.5).

The pole estimate error for the Ryugu images is 8.8069° . This error is an improvement over the ground based error of approximately 15° [88]. The average error around equatorial latitudes for the simulated images is less than 5° . The main contribution to this difference is the lower image resolution of the Hayabusa2 imagery. This results in fewer number of streaks that are smaller in

² Hayabusa2 Spice Kernels can be found [here](#)

Table 3.5: Pole Estimation Results on Hayabusa2 Imagery

Name	Estimated Value	True Value	Error
Eccentricity e	0.9880	0.9984	-0.0104
Orientation ϕ ($^\circ$)	94.6632	95.0518	-0.3886°
Pole Estimate (in the TIR camera frame)	$\begin{bmatrix} 0.0486 \\ -0.9868 \\ 0.1543 \end{bmatrix}$	$\begin{bmatrix} -0.0685 \\ -0.9961 \\ 0.0554 \end{bmatrix}$	8.8069°

length and therefore their ellipse fits are less accurate. As the body isn't symmetric, an offset between the CoBs of the IR images and the CoM of the body exists. This offset results in a blurrier stacked image and thus fewer streaks are extracted. In addition, the larger image spacing between images compared to the simulated cases also contributes to the increase in error (this is investigated in Section 3.7.2).

3.7 Algorithm Sensitivity Analysis

The algorithm results are sensitive to various tuning parameters (e.g. Table 3.2). The minimum length parameter (`minLength`) is designed to only admit streaks of a certain length so that small streaks that are not part of the latitude line ellipses do not affect the eccentricity and orientation estimate. Lowering this parameter would admit more data into the ellipse fitting and subsequent pole estimation procedure, but these smaller streaks are more susceptible to inaccurate ellipse fit. This is further exacerbated by pixel discretization of the streaks which strips some of the curvature information encoded within each streak that corresponds its elliptical nature. Thus, the ellipse fit is either not possible or has a large error. As the Ryugu image resolution is lower compared to the simulated Bennu and Itokawa images, this pixel discretization error has a larger effect on the ellipse fit. This contributes the degradation in performance between the simulated image sets of Bennu and Itokawa and the real Ryugu image set.

The match threshold parameter (`matchThresh`) sets the search radius within which the a particular streak is matched. On one hand, increasing this parameter will result in more incorrect streak matches. On the other hand, decreasing it may result in less matches overall which could

degrade performance. Therefore, tuning this parameter finds a balance between these two extremes.

The standard deviation used in the Canny edge detection algorithm (`cannySig`) is an additional parameter that affects the number of streaks extracted from the gradient image: the larger this parameter, the fewer number of streaks detected. With the results shown in Section 3.6, the general rationale for this parameter should be based on the apparent size of the body in the image. By this rationale, the standard deviation is reduced for the Ryugu images. The canny threshold parameter is set automatically based on the percentage of the non-edge pixels. In practice, we deem that this parameter doesn't affect the results significantly as the outlier rejection and minimum streak length parameters help to mitigate spurious edges.

As the algorithm is dependent on the contrast between latitude lines, performance will degrade when the imaged surface is at a uniform temperature. This can occur when the thermal inertia of the body is low and the axial tilt is close to 90° . With this illumination geometry, near-uniform temperatures are achieved on the northern and southern hemispheres of the body. In such cases, if the observer latitude is close to the pole ($\psi_{cam} \approx \pm 90^\circ$, i.e. $\phi \approx 0^\circ$), then fewer and weaker gradients are observed from the images. This impacts the quality of the extracted streaks and therefore impacts the pole estimate performance.

Nominally, the results on the simulated images utilize 201 images (N) spaced 1.8° (ΔI) apart across one spin period of the asteroid. In the remainder of this section, the sensitivity of the algorithm to the percentage of the spin period imaged and the spacing between images is investigated using the simulated images of Bennu. The analysis is conducted on the same set of axial tilt and observer latitude cases as the results in Section 3.6.

3.7.1 Algorithm Sensitivity to the Percentage of Spin Period Imaged

To investigate the sensitivity to the percentage of the rotational period spanned by the set of images, simulated images of Bennu taken over 50%, 65%, 75%, 85% are used in estimating the pole (at the same axial tilts and observer latitudes as in Section 3.6.1). The default image spacing shown in Table 3.2 is used here. Two different values for the canny edge detector standard deviation

parameter `cannysig` are used: the original parameter of 3 (from Table 3.2) and an adjusted value of 2.

Fig. 3.16 shows the pole estimate errors for both the original and the adjusted `cannySig` parameter in the form of histograms along with the mean and median errors. With the original tuning, the performance degrades in all of the cases with the degree of degradation increasing as fewer images are used. This is driven primarily by the increase in outliers compared to the baseline shown in Fig. 3.7b. These outliers occur due to the degradation in eccentricity performance at polar latitudes which is also observed in the baseline case (Fig. 3.8a) but appears to be worse in this section. As fewer images are used, the maximum error increases eventually reaching $\sim 25^\circ$ in the case shown in Fig. 3.16a for the 50% case. However, the median error stays around 5° which indicates that the performance doesn't degrade by much in a majority of the cases and the increase in the mean error is driven by more outliers as seen in the histograms in Fig. 3.16.

Using the adjusted parameter improves the results but with a diminishing margin of improvement as fewer images are used. This improvement is achieved by the reduction in the number of outliers with maximum error reduced to approximately 20° from 25° (in Fig. 3.16a). This analysis shows that the algorithm is sensitive to the number of images stacked. As fewer images are stacked, the gradients are less likely to correlate with the latitude lines which drives the degradation in pole estimates. However, it is also shown that by adjusting `cannySig`, the performance can be improved.

3.7.2 Algorithm Sensitivity to Image Spacing

In this section, three additional image spacing are investigated: spaced 0.9° , 2.7° , and 3.6° apart and taken over one spin period of the asteroid (at the same axial tilts and observer latitudes as in Section 3.6.1). For each of these cases, two different settings for the canny edge detection standard deviation parameters are used: the original parameter from Table 3.2 and an adjusted one shown in Table 3.6. The results are presented in Fig. 3.17 with the original and adjusted tuning in the form of histograms. The results with the original tuning show degradation in performance as the spacing between images increases. Similar to the results in Section 3.7.1, this performance

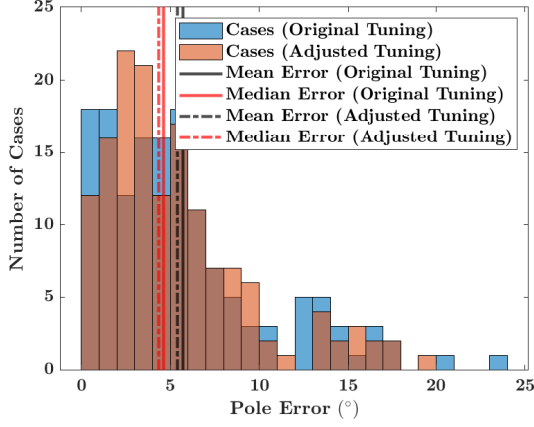
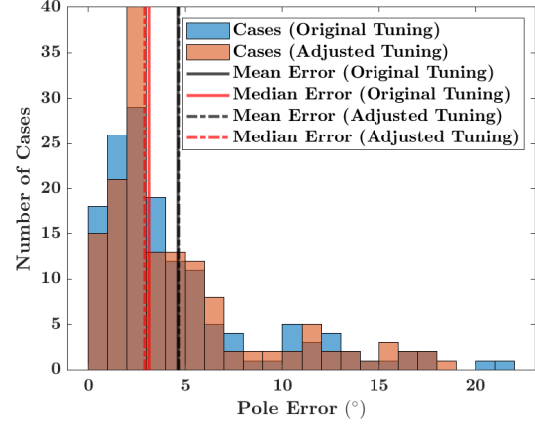
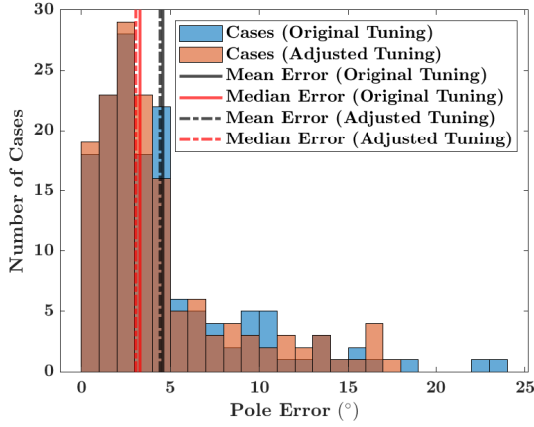
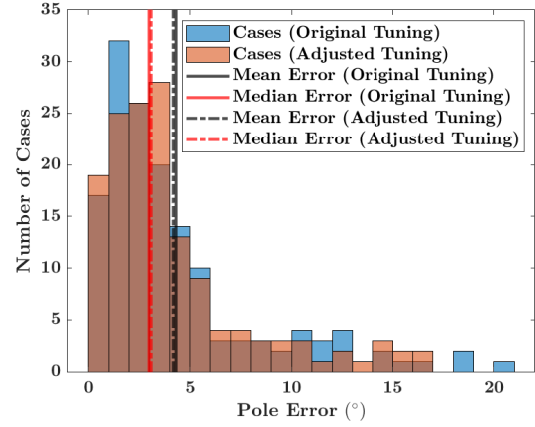
(a) 101 Images ($\sim 50\%$ of the spin period)(b) 131 Images ($\sim 65\%$ of the spin period)(c) 151 Images ($\sim 75\%$ of the spin period)(d) 171 Images ($\sim 85\%$ of the spin period)

Figure 3.16: Pole Error Histograms for Different Number of Bennu Images (Simulated Images)

Table 3.6: Adjusted `cannysig` Parameters for Image Spacing Study

Case	Adjusted Parameter Value
0.9°	2.5
2.7°	3.5
3.6°	4

degradation is driven by outliers. The results with the adjusted tuning parameters show improved performance. This improvement is primarily achieved by the reduction in the number of outliers.

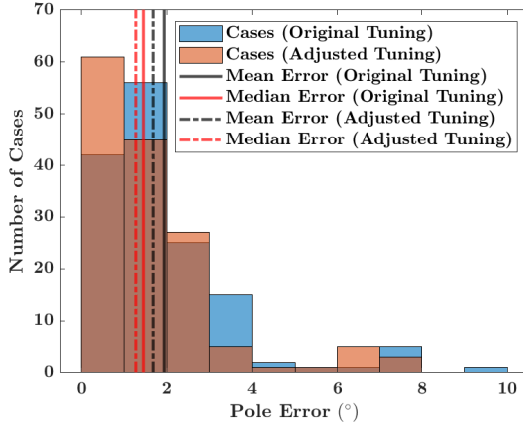
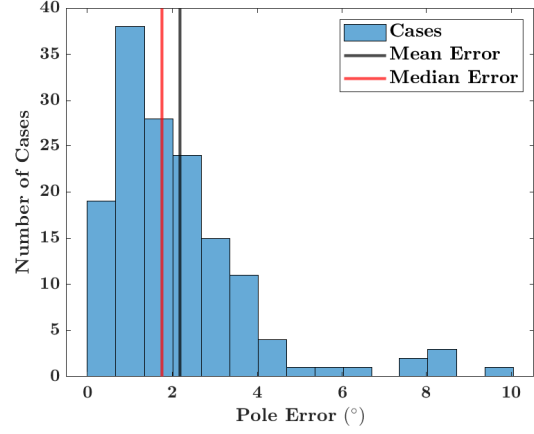
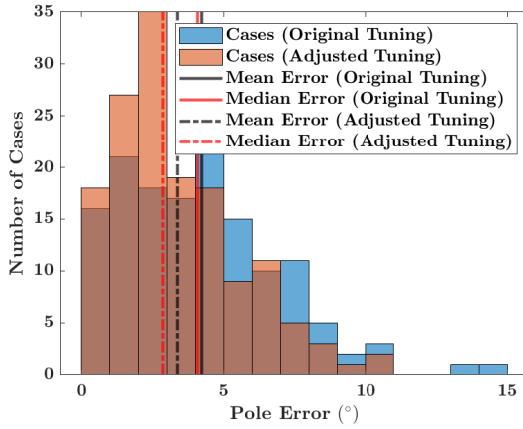
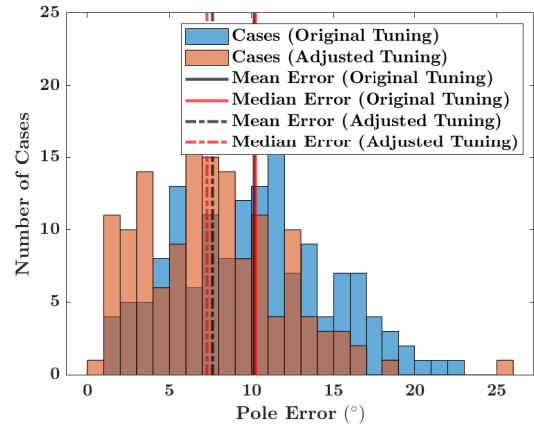
(a) Images Spaced 0.9° Apart(b) Images Spaced 1.8° Apart (baseline spacing)(c) Images Spaced 2.7° Apart(d) Images Spaced 3.6° Apart

Figure 3.17: Pole Error Histograms for Different Image Spacing on Simulated Bennu Images

3.8 Summary

In this chapter, a novel method of estimating the pole axis of an asteroid using on-board infrared imagery is presented and its performance is characterized. Infrared images are used due to their robustness to illumination geometries. The algorithm uses the appearance of latitude lines in a set of stacked images. These latitude lines are a result of the rotation of surface features on the body. The relative geometry between the pole axis and the camera frame is encoded in the appearance of the latitude lines in the images. Image processing steps are used to characterize

these latitude lines which includes the application of an ellipse fitting algorithm. A pole estimate is obtained by using the parameters of the estimated ellipses.

The performance of the algorithm is demonstrated on simulated infrared images of the asteroids Bennu and Itokawa across a range of illumination and observational geometries. The average pole estimate error for Bennu is 2.1853° which is an improvement compared to the ground-based pole estimate uncertainty of approximately 5° [91]. For Itokawa, the average pole estimate error is higher at approximately 6.5° due to its irregular shape. The simulated results illustrate the performance of the proposed algorithm at high phase angles ϕ ranging from 70° to 110° . While the results show the efficacy of using infrared images (as significant portions of the body remain in shadow at these phase angles), further analysis is warranted to fully characterize the algorithm performance in relation to solar phase angle.

The algorithm is also demonstrated on real infrared imagery using Hayabusa2's infrared images of asteroid Ryugu with a 8.8069° pole estimate error. This performance is affected by several factors: (1) comparatively fewer images (64 for Ryugu, 201 in the simulated cases on Bennu and Itokawa), (2) larger and non-uniform spacing between images ($\sim 5.635^\circ$ for Ryugu, 1.8° in the simulated cases), (3) lower image resolution (248×328 for Ryugu, 550×550 in the simulated cases), (4) non constant spacecraft relative position. The algorithm requires the images to be centered by the center of mass (CoM) but the Hayabusa2 images are centered by the image centers of brightness (CoBs). While the offset between CoB of an infrared image and the CoM of the body is minimal, the algorithm is still sensitive to this parameter. Estimating the CoM would reduce the blur in the stacked image which allows for more streaks to be extracted and thereby improve the pole estimate.

The algorithm's sensitivity to the spacing between images and the percentage of spin period imaged is characterized. With the default algorithm tuning parameters, the performance degrades as the image spacing increases or the percentage of spin period decreases. However, changing the edge detection standard deviation improves the results by reducing the number of outliers in the ellipse fit.

The pole estimation algorithm presented in this paper does not require a prior estimate of

the pole. It's intended to be used to obtain such an initial estimate using on-board imagery without incurring the computational and resource costs of ground based methods. As the results from the Hayabusa2 images show, this method is able to give a more accurate pole estimate compared to the extensive ground based study. These initial estimates can be used to initialize SPC or SPG shape and pole estimation methods to further refine the pole estimate.

Chapter 4

Binary Orbit Pole Estimation

4.1 Introduction

Binary asteroids comprise approximately 15% of the known near earth asteroid (NEA) population [78, 103] and have also been identified in the main belt [82]. Characterizing the mutual orbit of a binary system is of interest to the spaceflight and planetary science communities. Due to the sizes of these bodies and their distance from Earth, ground-based lightcurve observations have limited accuracy. In this chapter, an algorithm to estimate this orbit pole (orbital angular momentum vector) from spacecraft imagery is presented.

This mutual orbit, over short timescales, can be modeled by two-body gravitational dynamics and therefore, the orbit can be approximated by an ellipse. This orbit appears as an ellipse across a sequence of images (e.g., from a spacecraft camera assuming the barycenter is known). This observed ellipse over the set of images is a *projection* of the ellipse of the mutual orbit. By characterizing this projected ellipse, an estimate of the orbit pole can be obtained. While the method presented here can be applied to both optical and infrared images, infrared images are chosen for their robustness to illumination geometries.

4.2 Method

The orbit of a binary asteroid system can be modeled (over few orbital periods) using simplified two-body (Keplerian) dynamics. Thus, the motion of the secondary around the primary is approximated as an ellipse as shown in Fig. 4.1. Given an observer at some arbitrary distance from

the center of the orbit along $\mathcal{I}\hat{\mathbf{r}}_{cam} = \frac{\mathcal{I}\mathbf{r}_{cam}}{\|\mathcal{I}\mathbf{r}_{cam}\|}$, the orbit appears as an ellipse due to the projective geometry of conic sections. This projection shares some similarities with the projection of latitude lines from Chapter 3 with one difference: latitude lines are always circular where as binary orbits may be eccentric. Thus, in the latter case, the projection is a function of both the camera/orbit pole geometry but also the eccentricity of the binary orbit. In general, a given projected ellipse may not correspond to a unique combination of binary orbit eccentricity and camera/orbit pole geometry. However, it is possible to recover the orbit pole geometry if the binary orbit were circular.

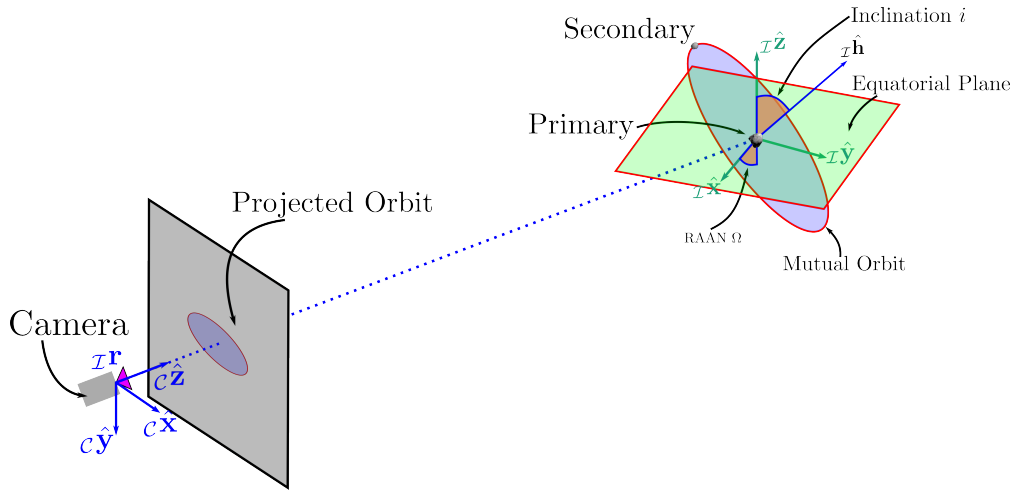


Figure 4.1: Geometry of a binary asteroid system modeled with Keplerian dynamics

Under this circular binary orbit assumption, the eccentricity of the projected ellipse is only a function of the observer latitude ψ_{cam} just as it is with latitudes lines (Chapter 3). Therefore, the eccentricity and orientation of the projected ellipse can be used to derive an estimate of the orbit pole. Consider a set of images in which the orbit of a secondary asteroid is observed (Fig. 4.2). Aligning the images about the primary (typically the bigger asteroid) and stacking them shows a trace of the projected orbit of the secondary. An example of this stacked image is shown in Fig. 4.2.

An ellipse fitting algorithm [36] is applied to this projected ellipse to extract the eccentricity e , major axis $c\hat{\mathbf{a}}$, minor axis $c\hat{\mathbf{b}}$. The major axis $c\hat{\mathbf{a}}$ lies in the camera plane and is estimated using the orientation of the ellipse in the camera plane ϕ (Eq. (4.1)). The ellipse fit, major and

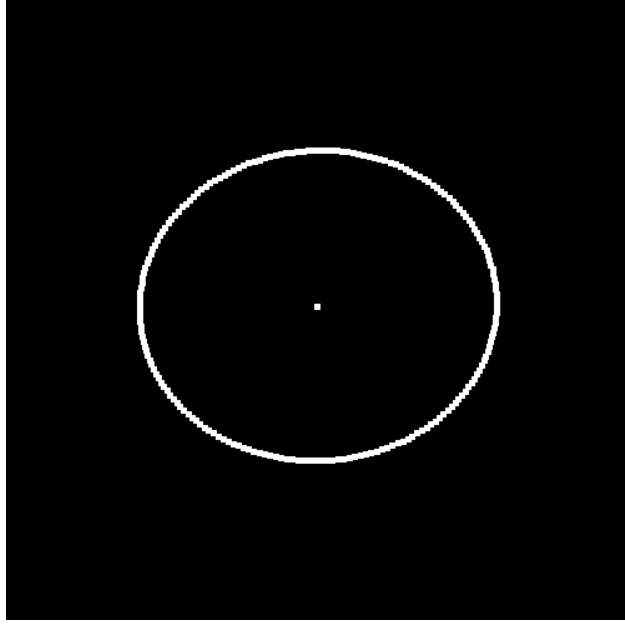


Figure 4.2: Stacked Image aligned around the primary shows the projected orbit

minor axes, and the orientation are shown in Fig. 4.3.

$${}_c\hat{\mathbf{a}} = \begin{bmatrix} \cos \phi \\ -\sin \phi \\ 0 \end{bmatrix} \quad (4.1)$$

Using the estimated eccentricity of the projected ellipse e , the angle between the orbit pole and the camera boresight β is given by:

$$\beta = \sin^{-1} e \quad (4.2)$$

The orbit pole estimate in the camera frame is estimated by rotating the camera boresight ${}_c\hat{\mathbf{z}}$ about the major axis ${}_c\hat{\mathbf{a}}$ by this angle β given by Eq. (4.2).

$${}_c\hat{\mathbf{h}} = \begin{bmatrix} \sin \phi \sin \beta \\ \cos \phi \sin \beta \\ -\cos \beta \end{bmatrix} \quad (4.3)$$

The orbit pole estimate in the inertial frame is obtained by using the known transformation between

the inertial and camera frames:

$${}_I\hat{\mathbf{h}} = {}_C^I\mathbf{R} {}_C\hat{\mathbf{h}} \quad (4.4)$$

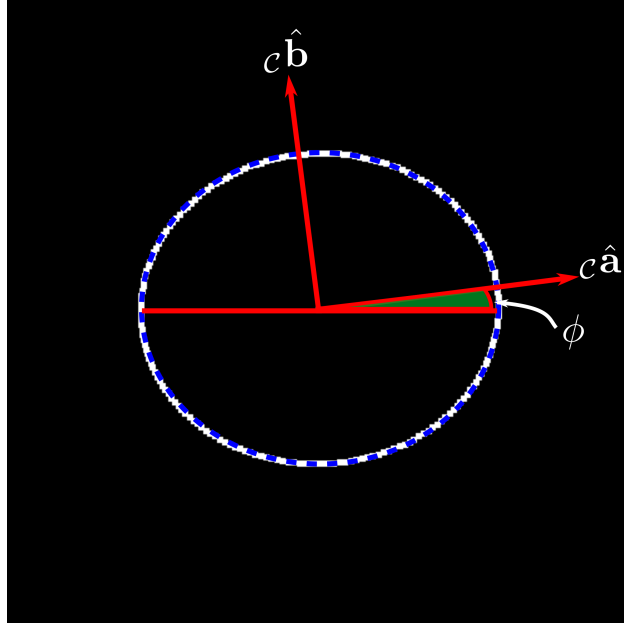


Figure 4.3: Projected ellipse fit (in blue) with orientation ϕ , major axis ${}_C\hat{\mathbf{a}}$ and minor axis ${}_C\hat{\mathbf{b}}$

4.3 Results

In this section, the method outlined in Section 4.2 is applied to simulated images of a binary system. The following is a list of assumptions of the results presented in this section.

- (1) Barycenter of the binary system is within the primary asteroid.
- (2) Camera boresight is pointed at the primary.
- (3) During the observational arc, the relative distance between the camera and the binary system doesn't change.

The semi-major axis of the simulated orbit is 5 km, the camera is at a distance of 1000 km from the primary and has a field-of-view of 1° with an image resolution of 500×500 pixels. As the

algorithm assumes that the binary orbit is circular (Section 4.2), the results presented here test that assumption in various observational geometries in which the following quantities are varied: the orbital inclination i , right ascension Ω , argument of periapsis ω , camera latitude ψ_{cam} , camera longitude α_{cam} . In each of these analyses, the eccentricity of the binary orbit (e) is also varied. In doing these parametric analyses, the circular binary orbit assumption is tested and validated. The parameters and the resulting orbit pole estimate errors are summarized in Table 4.1. In the results shown in Figs. 4.4 to 4.8, cases with smaller eccentricities had lower orbit pole error and larger errors are seen as the binary orbit eccentricity increases. This indicates that the circular orbit assumption is only valid for low eccentricities.

4.3.1 Eccentricity vs. Inclination

In this analysis, the orbital eccentricity and inclination are varied while the other parameters are held constant. These parameters are shown in the first row of Table 4.1. For each combination of eccentricity and inclination, images are simulated and the orbit pole is estimated. The plot shown in Fig. 4.4 illustrates the orbit pole error across the cases. For orbital eccentricities less than 0.3, the errors across all inclinations is less than 5° . Near 90° inclinations, the eccentricity does not have an effect on the orbit pole estimates with errors less than 5° . At these inclinations, the orbit pole is closely aligned with the boresight vector. In these cases, the orbital eccentricity only affects the camera plane component of the pole axis and thus has a minimal effect on the overall error. Further analysis is required to verify whether this generalizes across various observational and orbital geometries.

4.3.2 Eccentricity vs. Right Ascension

In this analysis, the orbital eccentricity and right ascension of the ascending node are varied while the other parameters are held constant (second row of Table 4.1). The plot shown in Fig. 4.5 illustrates the orbit pole error across the cases. The orbit pole is less sensitive to the right ascension compared to the inclination.

Table 4.1: Summary of parameters and results for the binary orbit pole study. For each case, orbital eccentricity and one other parameter are changed.

Case	Parameters	Orbit Pole Error Statistics
e vs. i (Section 4.3.1)	$e = 0, 0.010204, 0.020408, \dots, 0.4898, 0.5$	
	$i = 0^\circ, 10^\circ, 20^\circ, \dots, 180^\circ$	
	$\Omega = 70^\circ$	Mean Error: 3.2291°
	$\omega = 150^\circ$	Median Error: 1.2521°
	$\psi_{cam} = 70^\circ$	Std. Dev.: 4.7904°
	$\alpha_{cam} = 100^\circ$	
e vs. Ω (Section 4.3.2)	$e = 0, 0.010204, 0.020408, \dots, 0.4898, 0.5$	
	$i = 50^\circ$	
	$\Omega = 0^\circ, 10^\circ, 20^\circ, \dots, 360^\circ$	Mean Error: 2.1607°
	$\omega = 150^\circ$	Median Error: 1.2814°
	$\psi_{cam} = 70^\circ$	Std. Dev.: 2.4685°
	$\alpha_{cam} = 100^\circ$	
e vs. ω (Section 4.3.3)	$e = 0, 0.010204, 0.020408, \dots, 0.4898, 0.5$	
	$i = 50^\circ$	
	$\Omega = 70^\circ$	Mean Error: 1.4393°
	$\omega = 0^\circ, 10^\circ, 20^\circ, \dots, 360^\circ$	Median Error: 0.9972°
	$\psi_{cam} = 70^\circ$	Std. Dev.: 1.5868°
	$\alpha_{cam} = 100^\circ$	
e vs. ψ_{cam} (Section 4.3.4)	$e = 0, 0.010204, 0.020408, \dots, 0.4898, 0.5$	
	$i = 50^\circ$	
	$\Omega = 70^\circ$	Mean Error: 1.6966°
	$\omega = 150^\circ$	Median Error: 0.8125°
	$\psi_{cam} = -90^\circ, -80^\circ, \dots, 90^\circ$	Std. Dev.: 2.1228°
	$\alpha_{cam} = 100^\circ$	
e vs. α_{cam} (Section 4.3.5)	$e = 0, 0.010204, 0.020408, \dots, 0.4898, 0.5$	
	$i = 50^\circ$	
	$\Omega = 70^\circ$	Mean Error: 2.2372°
	$\omega = 150^\circ$	Median Error: 1.3456°
	$\psi_{cam} = 70^\circ$	Std. Dev.: 2.5302°
	$\alpha_{cam} = 0^\circ, 10^\circ, 20^\circ, \dots, 360^\circ$	

4.3.3 Eccentricity vs. Argument of Periapsis

In this study, the orbital eccentricity and argument of periapsis are varied while the other parameters are held constant (third row of Table 4.1). The plot shown in Fig. 4.6 illustrates the orbit pole error across the cases. Since the argument of periapsis is an angle within the plane of the orbit, it mostly changes the orientation of the projected ellipse in the image plane and this orientation is effectively estimated from the ellipse fitting algorithm. Thus, the orbit pole estimate

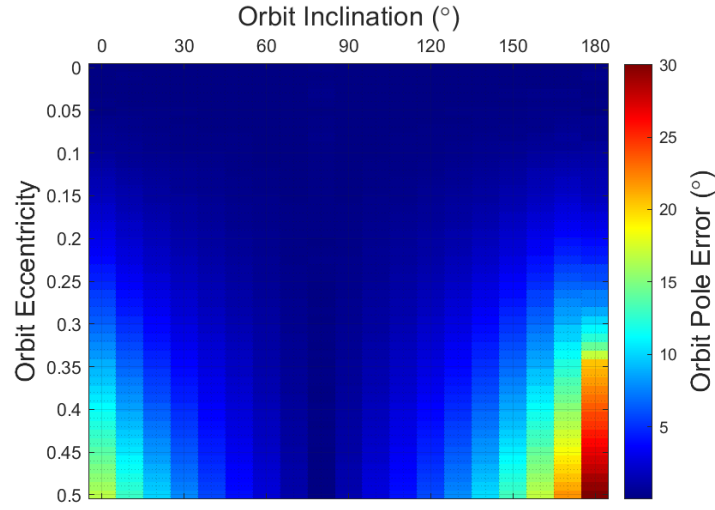


Figure 4.4: Orbit Pole Error across Different Orbital Eccentricities and Inclinations

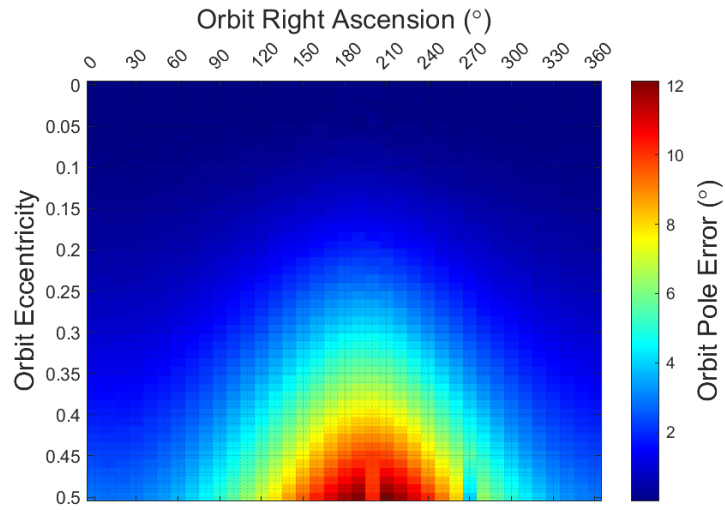


Figure 4.5: Orbit Pole Error across Different Orbital Eccentricities and Right Ascensions

is not effected by changing the argument of periapsis. While Fig. 4.6 shows this phenomenon, two outliers are present. The two outliers occur due to special geometry caused the coupling of the argument of periapsis and camera latitude. Further analysis is warranted to determine a trend exists across different camera latitudes and arguments of periapsis.

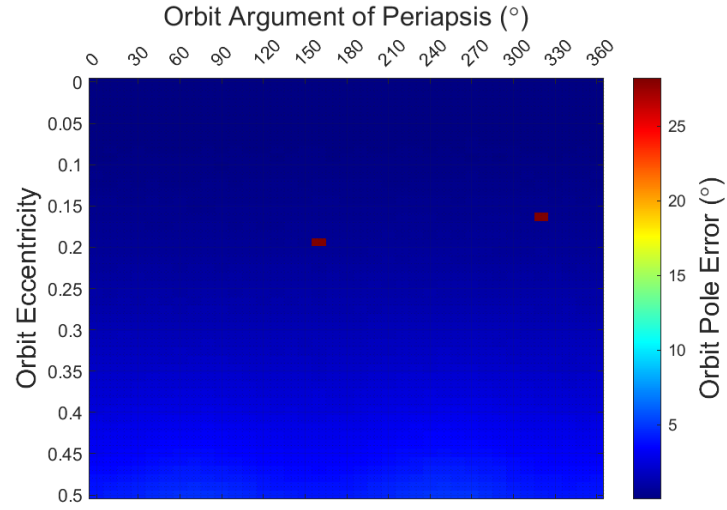


Figure 4.6: Orbit Pole Error across Different Orbital Eccentricities and Arguments of Periapsis

4.3.4 Eccentricity vs. Camera Latitude

The orbital eccentricity and camera latitude are varied while the other parameters are held constant (fourth row of Table 4.1). The plot shown in Fig. 4.7 illustrates the orbit pole error across the cases. Low eccentricity cases have better performance compared to the higher eccentricity cases. However, the orbit pole is less sensitive to the camera latitude compared to the inclination.

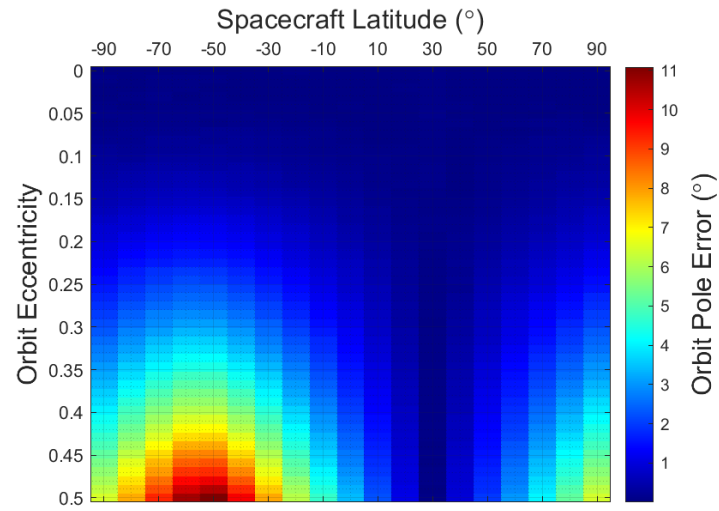


Figure 4.7: Orbit Pole Error across Different Orbital Eccentricities and Observer Latitudes

4.3.5 Eccentricity vs. Camera Longitude

The orbital eccentricity and camera longitude are varied while the other parameters are held constant (fifth row of Table 4.1). The plot shown in Fig. 4.8 illustrates the orbit pole error across the cases. Low eccentricity cases have better performance compared to the higher eccentricity cases. The sensitivity of the orbit pole to the camera longitude is similar to its camera latitude sensitivity.

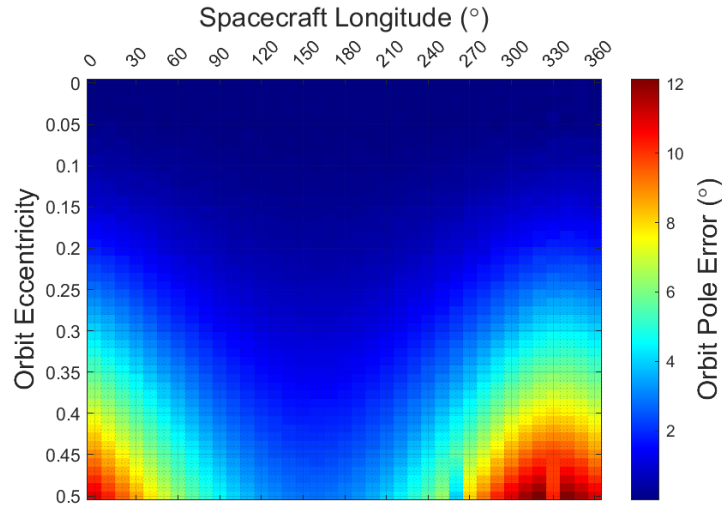


Figure 4.8: Orbit Pole Error across Different Orbital Eccentricities and Observer Longitudes

4.4 Summary

In this chapter, an algorithm to estimate the orbit pole of a binary asteroid system from camera observations of the projected orbit is introduced. This algorithm leverages the projective geometry of conic sections. Under an assumption that the mutual orbit is circular, the eccentricity and orientation of the projected ellipse are a function of the relative geometry between the camera and the orbit pole. The algorithm inverts this geometry to produce an estimate of the orbit pole. The algorithm is characterized in a set of parametric studies in which the orbital eccentricity and one other parameter is varied. These analyses show that the algorithm is most sensitive to orbital

inclination and least sensitive to the argument of periapsis. Across all of these studies, the orbit pole is estimated within 1° for orbital eccentricities of 0.2. As is shown in [102], most known binary orbital eccentricities are less than 0.2. This indicates that the algorithm can be applied to obtain accurate orbit pole estimates for a significant number of the binary asteroid systems.

Further analysis is necessary to characterize the algorithm's sensitivity to camera pointing errors, frequency and percentage of the orbit observed. In addition, the removal of the circular orbit assumption would improve the efficacy of the algorithm. It is observed that the deviation between the center of the projected orbit and the projected barycenter (in the image plane) is correlated to the orbital eccentricity. This may be used to estimate the orbital eccentricity which improves the estimate of the orbit pole (especially for higher eccentricities).

Chapter 5

Shape from Silhouette

5.1 Introduction

Once the pole axis is estimated from a set of infrared images (Chapter 3), the silhouettes (*i.e.* visual extents) from these images can be used to derive an initial shape model. Such Shape from Silhouette (SfS) algorithms are applicable once the small body is resolved (*i.e.* spans more than a pixel) until it spans the entire field-of-view (FOV) of the camera. The silhouette is a robust feature that can be extracted from resolved image of a small body while other feature descriptors (*e.g.* SIFT [73], SURF [18], ORB [110], KAZE [6] etc.) require the camera to be in close proximity of the small body. Preliminary iterations of the algorithm presented in this chapter are found in Refs. [14, 124]. The presented algorithm utilizes silhouettes from images over at least one spin period to infer the shape of the body. The resultant shape models are useful for relative navigation and to initialize SPC [43] or other shape refinement algorithms (Chapter 6). While the algorithm presented here can be applied to optical images (as demonstrated in [134]), infrared images offer robustness to illumination conditions. Therefore, the version of the SfS algorithm presented here (specifically the image processing steps) is intended to be used with infrared images. With minimal changes to the image processing, the algorithm can be modified to work for optical images.

5.1.1 Assumptions

The following is a list of assumptions of the algorithm presented in this chapter:

- (1) The relative position between the small body and the camera is known.

- (2) Small body pole axis orientation ${}_{\mathcal{I}}\hat{\mathbf{p}}$ is known. This implies that the small body to inertial frame rotation matrix ${}_{\mathcal{B}}^{\mathcal{I}}\mathbf{R}$ is known. The sensitivity of the algorithm to pole axis errors is characterized in Section 5.4.3.
- (3) Camera attitude relative to the inertial frame ${}_{\mathcal{C}}^{\mathcal{I}}\mathbf{R}$ is assumed to be perfect owing to the accuracy of star trackers.
- (4) An estimate of the maximum size of the body S is known.

5.2 Outline

The rest of this chapter is structured as follows. Section 5.3 outlines the proposed shape modeling algorithm including the image processing steps (Section 5.3.1). Results of the algorithm on synthetic and real images are presented in Section 5.4. Finally, a summary of this chapter is made in Section 5.5.

5.3 Algorithm

Given an image in which the entire body is within the camera’s field-of-view (FOV), the extent of the body is clearly visible even when no features within the body (*e.g* rocks, craters etc.) are discernible. This visible extent, *the silhouette*, represents the projection of the small body shape onto the camera’s image plane. In an image, the silhouette is parameterized by the set of pixels on the boundary of the body. Due to projective geometry, each *silhouette pixel* corresponds to a line of possible locations of the corresponding small body shape point. From different images of the body across time, different silhouette pixels and corresponding lines can be obtained. Each set of lines from one image constrains the shape from a two-dimensional, perspective. The sets of lines from multiple images are used to constrain the three-dimensional shape of the small body. The shape modeling algorithm presented here uses these lines to extract points corresponding to the small body shape.

The flowchart for the proposed algorithm is shown in Fig. 5.1. The algorithm steps are

divided into two sequential categories: *Image Processing* (Section 5.3.1) and *Point Cloud Extraction* (Section 5.3.2). Some algorithm steps can be executed as images are acquired (labeled *per image* and colored orange) while others are executed once all images are acquired (labeled *all images* and colored blue).

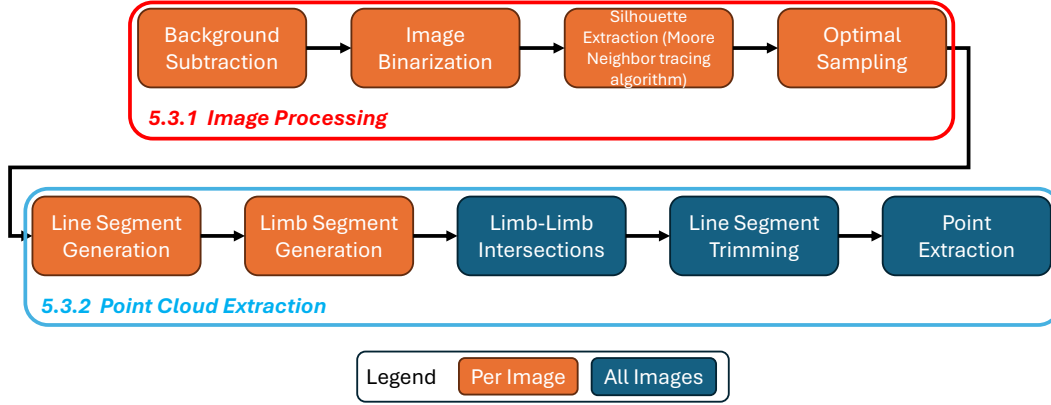


Figure 5.1: Shape from Silhouette (SfS) Algorithm Flowchart

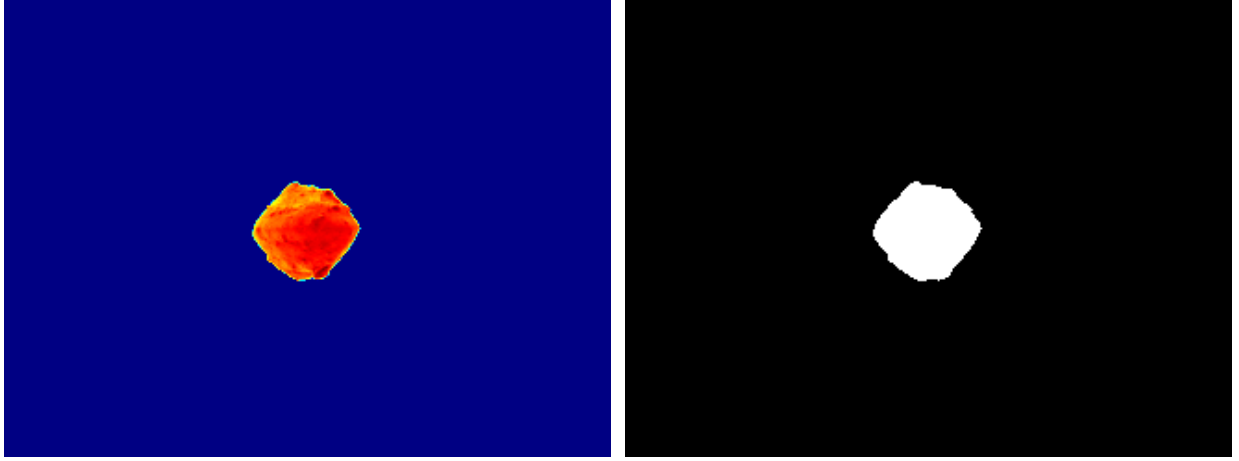
5.3.1 Image Processing

This section outlines image processing steps that are used to extract the silhouette information from an infrared image. In the case that optical images are used, an additional step is necessary after image binarization step to extract the illuminated limb from the silhouette (*i.e.* subtract the terminator). Details of this step and results of this algorithm on optical images are shown in Ref. [134].

5.3.1.1 Background Subtraction and Binarization

The objective of the image processing pipeline is to extract the silhouette from each image. The first step in this pipeline is to segment the body from the background. This can be achieved using a k-means clustering algorithm [12]. As the body is at a higher temperature compared to the background, the cluster with the larger median image intensity is identified as the body. As the difference in intensities between neighboring pixels does not affect the silhouette, the image

is binarized. An example of an infrared image and the output of the binarization process on this image are shown in Figs. 5.2a and 5.2b respectively.



(a) Example Infrared Image of Ryugu (redder spots correspond to higher temperatures) (b) Binary Image (body shown in white and background shown in black)

Figure 5.2: Example infrared image of Ryugu, and the output of the binarization process

5.3.1.2 Silhouette Extraction via Moore Neighborhood Tracing Algorithm

Silhouette pixels are extracted using a Moore neighborhood based tracing algorithm [81] and placed in the desired neighboring order. This order is necessary to construct the limb sections (Section 5.3.2.5) from a neighboring pair of pixels. A simpler sorting method employed in previous SfS algorithms [14] uses the angles of each pixel relative to the center of brightness (CoB). However, this method has been shown to not be robust to concave silhouettes like that of 67P/C-G [124] and results in poor shape modeling performance. Instead, the Moore Neighborhood Tracing algorithm is robust at extracting concave silhouettes (like that of 67P/C-G). While the simpler sorting method may be adequate for convex silhouettes (like those of Bennu), the Moore Neighborhood sorting algorithm is chosen here to maintain robustness regardless of the silhouette. A step-by-step illustration of this method is shown in Fig. 5.3.

The Moore neighborhood is defined as the set of pixels that share an edge or vertex with a given pixel (see Inset in Fig. 5.3). The tracing algorithm searches this neighborhood around the

current pixel (shown in pink in Fig. 5.3) in a clockwise or counterclockwise direction to identify the neighboring pixel (shown in orange in Fig. 5.3). The SfS algorithm has no direction preference for the silhouettes (clockwise or counterclockwise) as long as all the silhouettes are sorted in the same order. However, a clockwise order is used here. Pixels in the neighborhood that are already sorted (shown in blue in Fig. 5.3) are skipped in the search (as seen in the Inset). When the search returns to the starting pixel, if the search direction around the starting pixel is the same as the initial search direction, the algorithm terminates (*Jacob's Stopping Criteria*¹). Unlike simpler sorting methods such as angle based sorting, this method is effective for both convex and concave silhouettes [124].

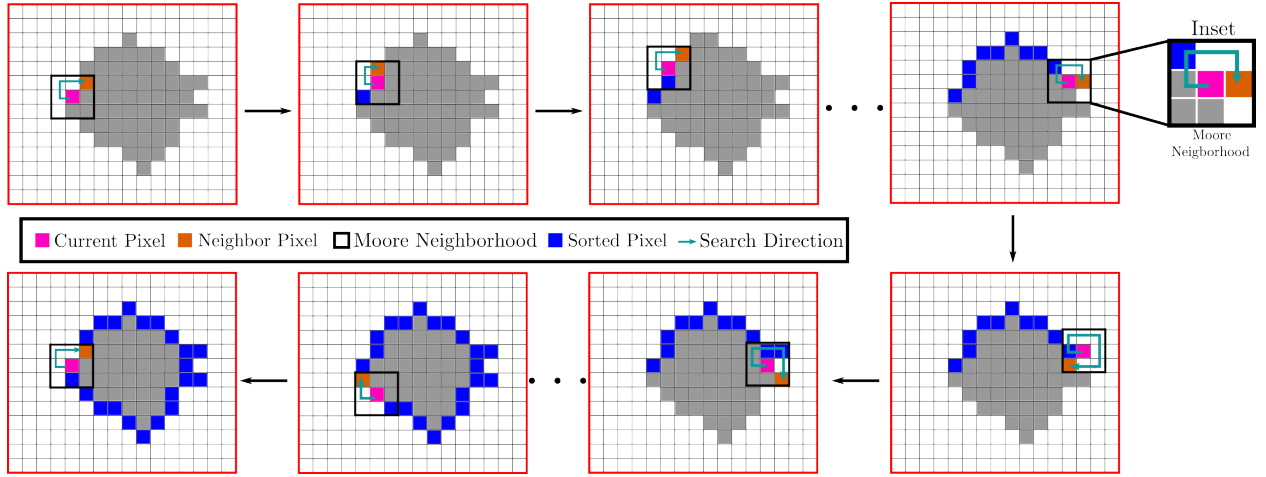


Figure 5.3: Illustration of the Moore neighborhood based Tracing Method. Each red box illustrates one iteration of the method. Bottom-left red box demonstrates Jacob's Stopping Criteria. Tracing Method highlighted for one pixel in the Inset

5.3.1.3 Optimal Sampling of Silhouettes

In this section, a sampling method based on the relative angles between neighboring pixels is presented. This method is used to distinguish salient and unnecessary pixels in a silhouette. The number of silhouette pixels in an image of a body is dependent on the camera resolution, shape of the body, and the body relative camera distance. Using all silhouette pixels may be too

¹ A description of which can be found [here](#)

computationally intractable and every pixel does not necessarily provide salient shape information. The method presented here balances computational cost and shape model accuracy by effective downsampling the salient set of pixels.

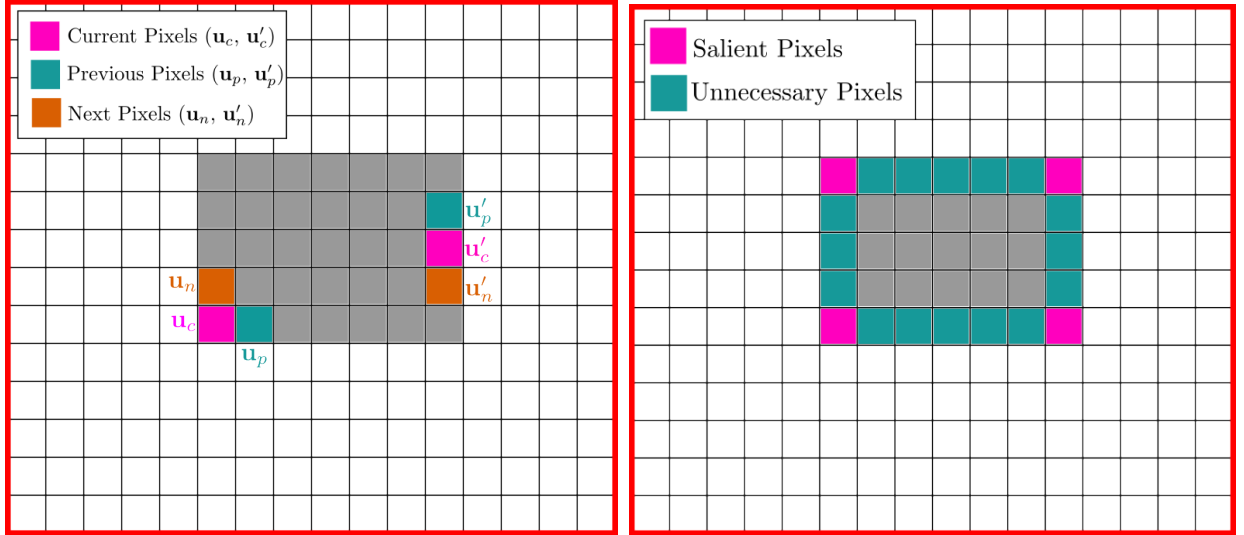
Consider the rectangle binary image shown in gray in Fig. 5.4a. Two sets of three pixels are shown in this figure: (1) $\mathbf{u}_c, \mathbf{u}_p, \mathbf{u}_n$ and (2) $\mathbf{u}'_c, \mathbf{u}'_p, \mathbf{u}'_n$. The subscript c denotes an arbitrary current pixel and the subscripts n and p are used to denote the neighboring pixels. Angles between the current pixels ($\mathbf{u}_c, \mathbf{u}'_c$) and their corresponding neighboring pixels are computed:

$$\alpha_p \equiv \angle(\mathbf{u}_c, \mathbf{u}_p) = \text{atan2}(u_{c,y} - u_{p,y}, u_{c,x} - u_{p,x}) \quad (5.1)$$

$$\alpha_n \equiv \angle(\mathbf{u}_n, \mathbf{u}_c) = \text{atan2}(u_{n,y} - u_{c,y}, u_{n,x} - u_{c,x}) \quad (5.2)$$

$$\alpha'_p \equiv \angle(\mathbf{u}'_c, \mathbf{u}'_p) = \text{atan2}(u'_{c,y} - u'_{p,y}, u'_{c,x} - u'_{p,x}) \quad (5.3)$$

$$\alpha'_n \equiv \angle(\mathbf{u}'_n, \mathbf{u}'_c) = \text{atan2}(u'_{n,y} - u'_{c,y}, u'_{n,x} - u'_{c,x}) \quad (5.4)$$



(a) Optimal sampling method illustration for salient and unnecessary pixels ($\mathbf{u}_c$, and $\mathbf{u}'_c$ respectively) (b) Optimal Sampling Result (corners of the rectangle are salient and others are unnecessary)

Figure 5.4: Illustration of the Optimal Sampling Algorithm

For the pixels shown in Fig. 5.4a, it is evident that (1) $\alpha_p \neq \alpha_n$ and (2) $\alpha'_p = \alpha'_n$. The first result is used to deem that the pixel $\mathbf{u}_c$ is *salient* as it corresponds to a change in shape. The

second result shows that pixel $\mathbf{u}'_c$ is *unnecessary* as there is no angle change and hence no *unique* shape information. The result of applying this method on the rectangle is shown in Fig. 5.4b where the corners of the rectangles are marked *salient* (in pink) and the others are marked *invalid* (in green). From inspection, it is evident that only the four corner pixels of the rectangle are necessary to precisely describe the shape.

5.3.2 Point Cloud Generation

5.3.2.1 Generate Lines from Silhouette Pixels

As the silhouette pixels are projected from the body onto the image plane, they lie along a line from the optical center of the camera to the corresponding pixel (shown in Fig. 5.5). This *line-of-sight* ${}_c\ell$, corresponding to a pixel $\mathbf{u}$, is computed using the inverse of the camera calibration matrix $\mathbf{K}^{-1}$:

$$\mathbf{K}^{-1} = \begin{bmatrix} \frac{1}{d_x} & 0 & \frac{-s_x}{d_x} \\ 0 & \frac{1}{d_y} & \frac{-s_y}{d_y} \\ 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

$$d_x = \frac{f}{\mu_x} \quad (5.6)$$

$$d_y = \frac{f}{\mu_y} \quad (5.7)$$

$$\mathbf{u} = \begin{bmatrix} u_x \\ v_y \end{bmatrix} \quad (5.8)$$

$${}_c\ell = \mathbf{K}^{-1} \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} \quad (5.9)$$

s_x, s_y are the principal points of the camera, μ_x, μ_y are the pixel pitch, and f is the focal length.

Any of the infinitely many points along this line-of-sight ${}_c\ell$ could be the shape point(s) on the body that project onto the image plane at the pixel $\mathbf{u}$. The known camera-to-body distance D and maximum size of the body S (Assumptions 1 & 4 in Section 5.1.1) are used to constrain the possible shape point location(s) to a finite *line segment* $\mathbf{L}$. The start and end coordinates of $\mathbf{L}$ (${}_c\mathbf{r}_s$,

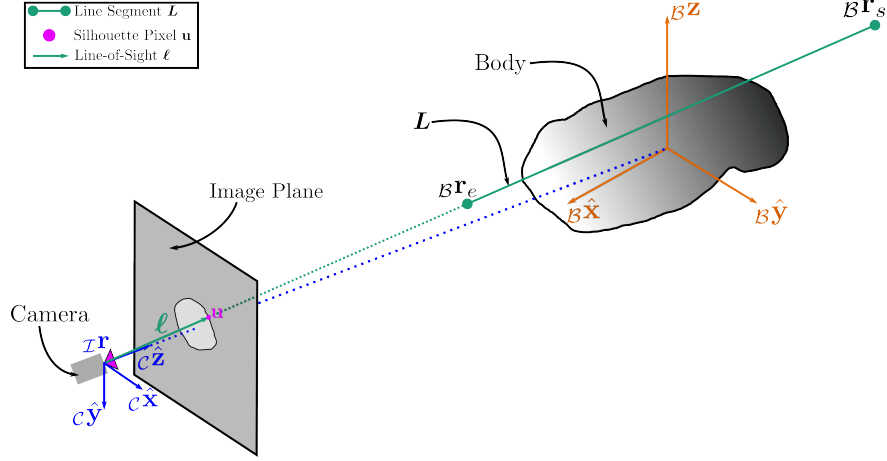


Figure 5.5: Silhouette pixel $\mathbf{u}$, line-of-sight ℓ , and line segment $\mathbf{L}$ with coordinates ${}_{\mathcal{B}}\mathbf{r}_s$ and ${}_{\mathcal{B}}\mathbf{r}_e$.

${}_{\mathcal{C}}\mathbf{r}_e$) are defined in the camera frame $\mathcal{C}$ by Eqs. (5.10) and (5.11) respectively. The corresponding shape point(s) are in the $-\ell$ direction from ${}_{\mathcal{C}}\mathbf{r}_s$ and $+\ell$ direction from ${}_{\mathcal{C}}\mathbf{r}_e$.

$${}_{\mathcal{C}}\mathbf{r}_s = {}_{\mathcal{C}}\ell(D + S) - {}_{\mathcal{C}}\mathbf{r} \quad (5.10)$$

$${}_{\mathcal{C}}\mathbf{r}_e = {}_{\mathcal{C}}\ell(D - S) - {}_{\mathcal{C}}\mathbf{r} \quad (5.11)$$

$${}_{\mathcal{C}}\mathbf{r} = {}_{\mathcal{I}}^{\mathcal{C}}\mathbf{R} {}_{\mathcal{I}}\mathbf{r} \quad (5.12)$$

The vector ${}_{\mathcal{C}}\mathbf{r}$ is the relative position of the body with respect to the camera expressed in the camera frame. An example of a line segment $\mathbf{L}$ is shown in Fig. 5.5. Equations (5.13) and (5.14) express the coordinates of $\mathbf{L}$ in the body frame $\mathcal{B}$. A line segment corresponding to each silhouette pixel in an image is generated using the aforementioned process.

$${}_{\mathcal{B}}\mathbf{r}_s = {}_{\mathcal{C}}^{\mathcal{B}}\mathbf{R} {}_{\mathcal{C}}\mathbf{r}_s \quad (5.13)$$

$${}_{\mathcal{B}}\mathbf{r}_e = {}_{\mathcal{C}}^{\mathcal{B}}\mathbf{R} {}_{\mathcal{C}}\mathbf{r}_e \quad (5.14)$$

5.3.2.2 Generate Limb Sections

A limb section $\mathbf{Q}$ is a quadrilateral consisting of the start and end coordinates of a neighboring pair of line segments in an image. An outward pointing normal ${}_{\mathcal{B}}\hat{\mathbf{n}}$ of $\mathbf{Q}$ is computed using Eq. (5.15). As $\mathbf{Q}$ corresponds to a segment on the body's silhouette, the shape of the body is

constrained to be on the $-\mathcal{B}\hat{\mathbf{n}}$ side of $\mathcal{Q}$ (*i.e.* the inward normal side). Two such neighboring limbs from image i , $\mathcal{Q}^{(i,1)}$ & $\mathcal{Q}^{(i,2)}$, are shown in Fig. 5.6 along with their constituent line segments $\mathcal{L}^{(i,1)}$, $\mathcal{L}^{(i,2)}$, and $\mathcal{L}^{(i,3)}$.

$$\mathcal{B}\hat{\mathbf{n}} = \frac{(\mathcal{B}\mathbf{r}_e^{(i,1)} - \mathcal{B}\mathbf{r}_s^{(i,1)}) \times (\mathcal{B}\mathbf{r}_s^{(i,2)} - \mathcal{B}\mathbf{r}_s^{(i,1)})}{\|(\mathcal{B}\mathbf{r}_e^{(i,1)} - \mathcal{B}\mathbf{r}_s^{(i,1)}) \times (\mathcal{B}\mathbf{r}_s^{(i,2)} - \mathcal{B}\mathbf{r}_s^{(i,1)})\|} \quad (5.15)$$

Equation (5.15) produces an outward pointing normal only when the silhouette points are sorted in a clockwise order (the order chosen in this chapter). Alternative form of Eq. (5.15) can be devised for a counterclockwise ordering of pixels.

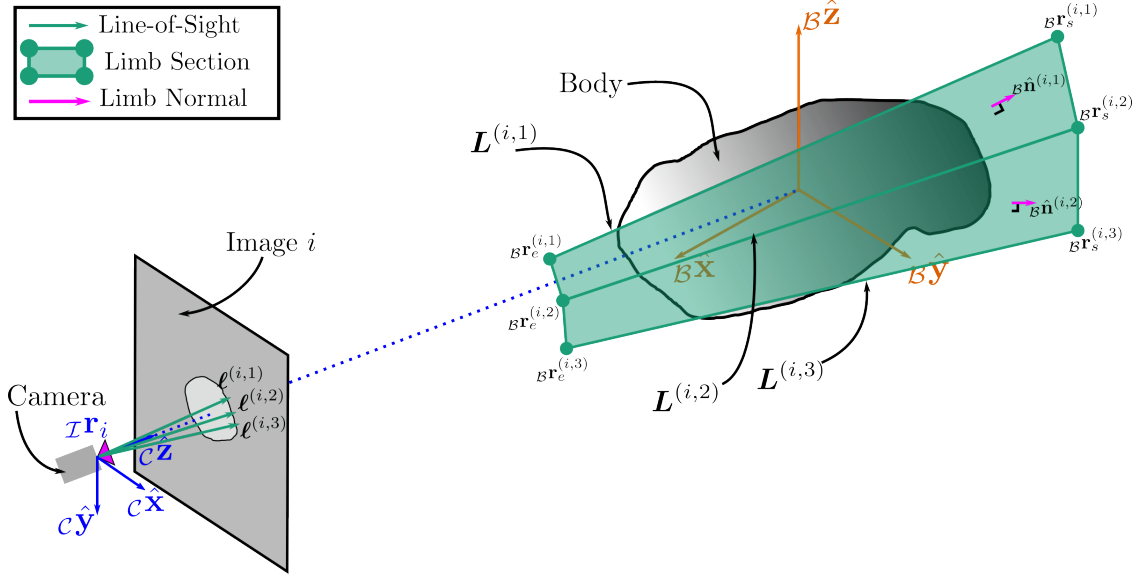


Figure 5.6: Neighboring lines-of-sight ($\ell^{(i,1)}$, $\ell^{(i,2)}$, $\ell^{(i,3)}$) from image i , corresponding line segments ($\mathcal{L}^{(i,1)}$, $\mathcal{L}^{(i,2)}$, and $\mathcal{L}^{(i,3)}$), limb sections $\mathcal{Q}^{(i,1)}$, $\mathcal{Q}^{(i,2)}$ (shaded green), and limb normals $\mathcal{B}\hat{\mathbf{n}}^{(i,1)}$, $\mathcal{B}\hat{\mathbf{n}}^{(i,2)}$

5.3.2.3 Limb-Limb Intersections

A polygon-polygon intersection algorithm (see Appendix B) is used to compute intersections between any two limb sections. In Fig. 5.7, two limb sections each from two images i and j are shown along with the intersecting points. Consider the intersection point $\mathcal{B}\mathbf{i}_s^{(1)}$ shown in the right side Fig. 5.7 which is closest to the line segment with coordinates $\mathcal{B}\mathbf{r}_e^{(i,1)}$, $\mathcal{B}\mathbf{r}_s^{(i,1)}$. Since this line segment is on the limb section from image i , the normal vector of the *intersecting* limb section from

image j is projected onto this line (using Eq. (5.16)). This component $n_{\parallel}$, shown in the Fig. 5.7, indicates that the shape point lies to the left of the intersection point $\mathcal{B}\mathbf{i}_s^{(1)}$. This process is repeated for all identified intersection points between two limbs. Note that intersection procedure is omitted for limb sections from the *same* image.

$$n_{\parallel} = \text{sign}(\mathcal{B}\hat{\mathbf{r}} \cdot \mathcal{B}\hat{\mathbf{n}}_{int}) \quad (5.16)$$

$$\mathcal{B}\hat{\mathbf{r}} = \frac{\mathcal{B}\mathbf{r}_e - \mathcal{B}\mathbf{r}_s}{\|\mathcal{B}\mathbf{r}_e - \mathcal{B}\mathbf{r}_s\|} \quad (5.17)$$

The normal vector $\hat{\mathbf{n}}$ at each intersection point of a line segment $\mathbf{L}$ is the average of the normals

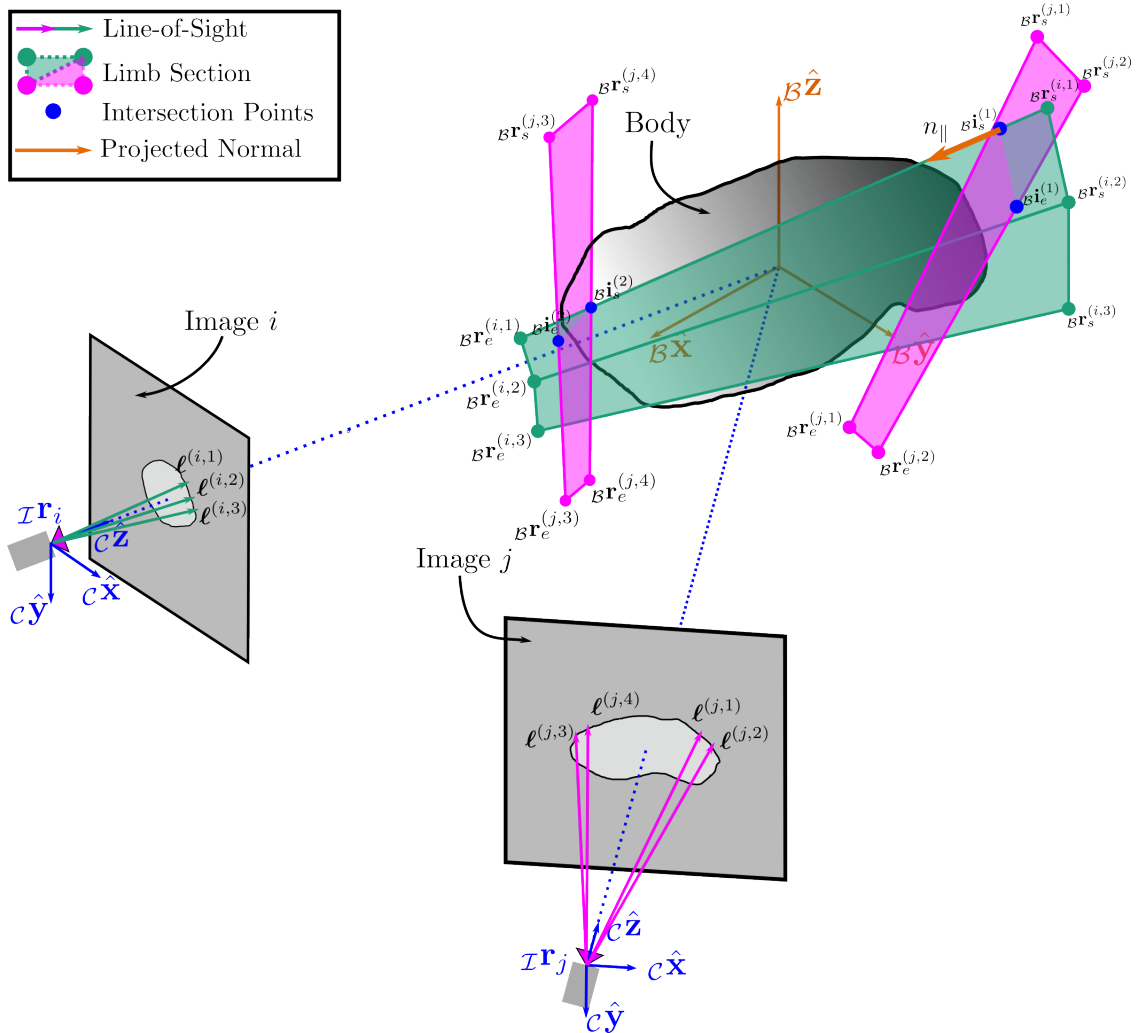


Figure 5.7: Demonstrating intersection ($\mathcal{B}\mathbf{i}_s$, $\mathcal{B}\mathbf{i}_e$) between limbs from image i and image j ($i \neq j$)

of each of the corresponding limbs that produced the intersection. The normal vector is computed

to facilitate the trimming process and also allows for the computation of *oriented* shape points, *i.e.* shape points with associated normal directions which indicate the surface slope. This information is necessary for some shape reconstruction algorithms [63] as discussed in Section 5.3.3.

5.3.2.4 Line Segment Trimming

For a given line segment $\mathbf{L}$ on image i , multiple intersections with limb sections ($\mathbf{Q}^{(j,1)}, \mathbf{Q}^{(j,2)}, \dots$) from another image j exist. For each intersection point (shown in blue in the inset of Fig. 5.8), a sign of the projected normal $n_{\parallel}$ is computed (Eq. (5.16)). The intersection points where the $n_{\parallel}$ changes is indicative of the section of this line segment $\mathbf{L}$ where the shape point lies. Consider the four intersection points shown on the line segment in Fig. 5.8 along with the corresponding projected normals $n_{\parallel}$. The *change* in the sign of $n_{\parallel}$ occurs between $\mathbf{bi}^{(2)}$ and $\mathbf{bi}^{(3)}$ indicating that the shape point lies between these two points. The segment between these two points is identified as the trimmed segment $\mathbf{T}^{(i,1)}$. The normal vector $\mathbf{b}\hat{\mathbf{n}}_s$ at a shape point $\mathbf{bs}$ is computed by averaging the normals corresponding to the two end points of the overlapping segment. This normal vector information is necessary for some shape reconstruction algorithms [63].

5.3.2.5 Point Extraction

The procedure outlined in the previous section results in one or more trimmed segment(s) $\mathbf{T}$ for a given line segment. For each intersecting image, several trimmed segments can be identified which can occur with non-convex bodies (Fig. 5.9a). An *overlapping segment* between all trimmed segments is obtained using the procedure illustrated in Fig. 5.9a. The start of the overlapping segment is the start coordinate of the trimmed segment with the *largest* distance (Eq. (5.18)) relative to the start of the line segment. The end of the overlapping segment is the end coordinate of the trimmed segment with the *shortest* distance (Eq. (5.19)) relative to the start of the line segment. If the start coordinate distance is *larger* than the end coordinate, the overlapping segment is *invalid* and no shape point corresponding to the given line segment is found. Otherwise, the midpoint of the overlapping segment $\mathbf{S}$ is identified as the shape point $\mathbf{bs}$ corresponding to given line segment

L . Equation (5.18) shows the distance between the start of the line segment $\mathcal{B}\mathbf{r}_s$ and the start of the trimmed segment $\mathcal{B}\mathbf{t}_s$ and Eq. (5.19) shows the distance between the start of the line segment and the end of the trimmed segment $\mathcal{B}\mathbf{t}_e$.

$$d_s = \|\mathcal{B}\mathbf{t}_s - \mathcal{B}\mathbf{r}_s\| \quad (5.18)$$

$$d_e = \|\mathcal{B}\mathbf{t}_e - \mathcal{B}\mathbf{r}_s\| \quad (5.19)$$

For concave bodies, more than one overlapping segment corresponding to L can be found. In this case, the midpoints of each overlapping segment are identified as shape points ($\mathcal{B}\mathbf{s}^{(1)}$, $\mathcal{B}\mathbf{s}^{(2)}$ shown in Fig. 5.9). As noted in Section 5.3.2.3, the normal of each intersecting point on a line segment L is computed from the average of the two corresponding limbs. The normal vector $\mathcal{B}\hat{\mathbf{n}}_s$ at a shape point $\mathcal{B}\mathbf{s}$ is computed by averaging the normals corresponding to the two end points of the overlapping segment. This normal vector information is necessary for some shape reconstruction algorithms [63].

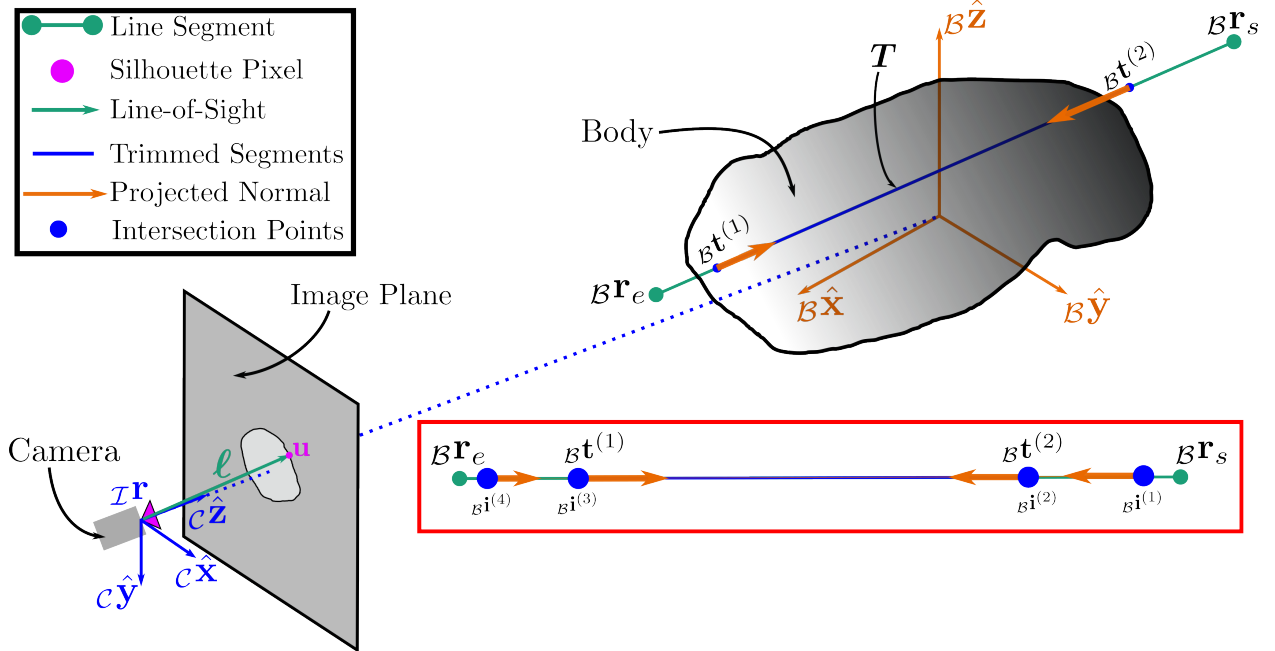
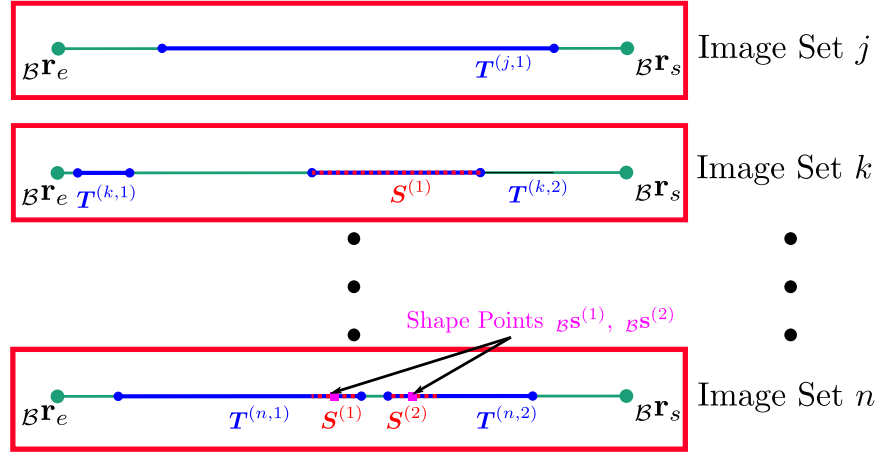
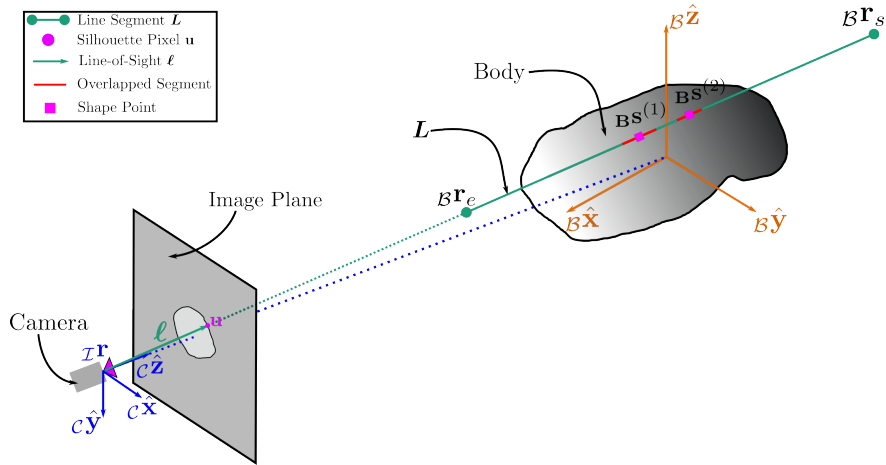


Figure 5.8: Illustration of the trimming procedure



(a) Procedure to obtain shape point(s) on a line segment L by overlapping trimmed segments T



(b) Extracted shape points $B s^{(1)}, B s^{(2)}$ corresponding to Line segment L

Figure 5.9: Illustration of Shape Point Extraction by overlapping trimmed segments T

5.3.3 Shape Reconstruction

With the algorithm steps outlined in the preceding sections, a point cloud representing the small body shape is obtained along with estimates of the normal vectors (an *oriented point set*). Often, it is useful to have a mesh (points with faces) instead of the points alone (*e.g.* for on-board image generation). The Screened Poisson Surface Reconstruction algorithm [63, 64] obtains a mesh from the oriented point clouds. Other mesh generation algorithms such as the Power Crust Algorithm [8] are available which don't require oriented point sets and preserves the size of the

input point cloud (*i.e.* size of mesh vertices equals size of point cloud). Either of these algorithms are superior to simpler counterparts such as Delaunay triangulation, especially for concave shapes. The results presented here (Section 5.4) are in the form of point clouds such that the SfS algorithm performance can be isolated from the influence of shape reconstruction algorithms.

5.3.4 SfS Algorithm

In this section, a pseudocode for the SfS algorithm is presented (Algorithm 3). The algorithm outputs shape points $\mathcal{B}\mathbf{s}$ and requires the following inputs: (1) distance between the camera and the body D , (2) approximate size of the body S , (3) inverse of the camera calibration matrix K^{-1} , and (4) vector position between small body and camera $\mathcal{I}\mathbf{r}$ ($D = \|\mathcal{I}\mathbf{r}\|$). Lines 2–14 are executed for each image independently and the rest of the algorithm has to be executed once all images are acquired (as illustrated in Fig. 5.1). Lines 1–5 illustrate the image processing steps necessary to extract the silhouette pixels from a sequence of images. Lines 6–14 outline the process of generating limb sections $\mathcal{Q}$ from each pair of neighboring line segments $\mathcal{L}$ from the same image. Lines 16–22 outline the limb intersection process between two limbs from different images. Duplicate intersections are avoided by only performing intersections between a limb $\mathcal{Q}_1$ from one image and another limb $\mathcal{Q}_2$ from a different image acquired *after* the first. This leads to improved computational efficiency (by a factor of 2) while not affecting algorithm performance. Lines 23–28 illustrate the process of trimming each line segments based on intersections with limb sections and subsequently obtaining a shape point $\mathcal{B}\mathbf{s}$ from overlapping the trimmed segments $\mathcal{T}$.

The computational complexity of the algorithm is $\mathcal{O}(MN)$ where M is the number of images and N is the number of pixels per image. The sub-sampling algorithm (Section 5.3.1.3) reduces the total number of pixels and avoiding duplicate intersections reduces the number of intersections that need to be computed. Together, these two steps reduce the overall computational cost of the algorithm.

Algorithm 3 SfS Algorithm

Input: $D, S, K^{-1}, \mathcal{I}\mathbf{r}$ **Output:** $_{BS}$

```

1: for each image  $I$  in  $N$  images do                                ▷ Process each image
2:    $B \leftarrow \text{removeBackground}(I)$                                 ▷ Section 5.3.1.1
3:    $BW \leftarrow \text{binarize}(B)$                                         ▷ Section 5.3.1.1
4:    $[U_s, V_s] \leftarrow \text{mooreNeighbor}(BW)$                         ▷ Section 5.3.1.2
5:    $[U_{ss}, V_{ss}] \leftarrow \text{optimalSS}(U_s, V_s)$                 ▷ Section 5.3.1.3
6:   for each pixel  $[u, v]$  in  $[U_{ss}, V_{ss}]$  do
7:      $\ell \leftarrow \text{generateLOS}([u, v], K^{-1})$                     ▷ Eq. (5.9)
8:      $[_B\mathbf{r}_s, _B\mathbf{r}_e] \leftarrow \text{generateLineSeg}(\ell, D, S, \mathcal{I}\mathbf{r})$     ▷ Eqs. (5.10), (5.11), (5.13) and (5.14)
9:      $L \leftarrow [_B\mathbf{r}_s, _B\mathbf{r}_e]$                                     ▷ Line segment
10:  end for
11:  for each neighboring pair  $L_1, L_2$  do                            ▷ Generate limb sections
12:     $Q \leftarrow [L_1, L_2]$                                         ▷ limb section
13:     $_{B}\hat{\mathbf{n}} \leftarrow \text{computeLimbNormal}(Q)$                     ▷ Outward Normal (Eq. (5.15))
14:  end for
15: end for
16: for  $Q_1$  and  $Q_2$  in limb sections do                                ▷ Section 5.3.2.3
17:   if  $\text{imageIndex}(Q_1) > \text{imageIndex}(Q_2)$  then
18:      $_{B}\mathbf{i} \leftarrow \text{intersectLimbs}(Q_1, Q_2)$                                 ▷ C
19:      $L_{int} \leftarrow \text{identifyLineSeg}(_B\mathbf{i}, Q_1, Q_2)$                 ▷ Line segment for  $_{B}\mathbf{i}$ 
20:      $n_{\perp} \leftarrow \text{signProjNorm}(L_{int}, _B\hat{\mathbf{n}})$                 ▷ Eq. (5.16)
21:   end if
22: end for
23: for each line segment do                                            ▷ Section 5.3.2.4
24:   for each intersecting image set do
25:      $T \leftarrow \text{trimSegment}(L, _B\mathbf{i}, n_{\perp})$                     ▷ Based on intersections
26:   end for
27:    $S \leftarrow \text{overlapSegments}(T)$                                 ▷ Section 5.3.2.5
28:    $_{BS} \leftarrow \text{getShapePoints}(S)$                                 ▷ Section 5.3.2.5
29: end for

```

5.4 Results

5.4.1 Shape Models Derived from Synthetic Images

In this section, synthetic images of Bennu, Itokawa, 67P/C-G, and Eros are used to generate their respective shape models. For each of the four small bodies, two sets of synthetic images are generated at two approach latitudes: 0° and 45° . Each set of images contains 81 images over one spin period of the body such that the body spins 4.5° about its pole axis between images. In the equatorial latitude (0°) images, the silhouettes accurately capture shape information across all latitudes. As it may not be feasible to approach the body along an equatorial latitude, it is useful to investigate expected shape modeling performance from a non-equatorial approach. Thus, a shape model derived from 45° latitude aims to illustrate the expected performance from a non-equatorial latitude. Table 5.1 contains statistics of the 0° and 45° shape models for Bennu, Itokawa, 67P/C-G, and Eros.

Small Body	Latitude ($^\circ$)	No. Points	RMS Error (m)	RMS Error (%)	Result
Bennu	0	15,746	1.333	0.4609	Fig. 5.10
	45	33,591	3.159	1.091	Fig. 5.11
Itokawa	0	9,608	4.056	1.303	Fig. 5.12
	45	19,947	4.672	1.500	Fig. 5.13
67P/C-G	0	16,832	53.99	2.038	Fig. 5.14
	45	30,380	54.27	2.049	Fig. 5.15
Eros	0	8,216	341.6	1.938	Fig. 5.16
	45	20,219	217.1	1.231	Fig. 5.17

Table 5.1: Statistics of shape models at 0° and 45° viewpoints. Rightmost column provides a reference to the figure showing the shape model performance

5.4.1.1 Bennu

A 200,000 facet reference shape model [17] is used to simulate the synthetic images of asteroid Bennu. The images capture Bennu over its spin period of approximately 4.3 hrs [68] and thus each image is spaced approximately 3.22 min apart. Characterization of shape models generated using

images from 0° and 45° latitudes are shown in Figs. 5.10 and 5.11 respectively. The equatorial images based shape model contains 15,746 points (Fig. 5.10a) and shows similar performance across latitudes with an overall RMS error of 1.333 m (0.4609% of the body radius). The shape model using images from a 45° latitude contains 33,591 points (Fig. 5.11a) and has an overall RMS error of 3.159 m (1.091% of the body radius). The higher RMS error is due the inaccurate shape characterization at latitudes north of 45° and south of -45° which are illustrated in the latitude-longitude plot in Fig. 5.11b.

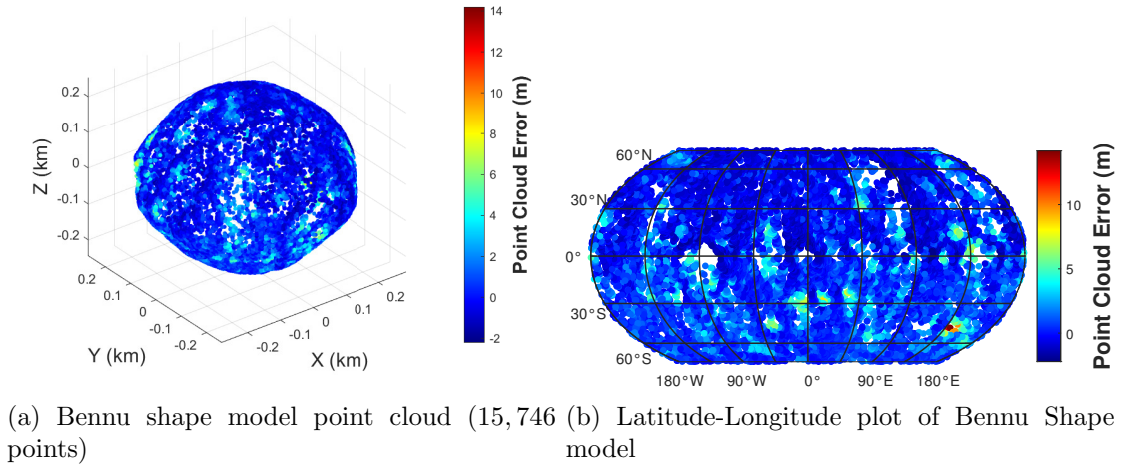


Figure 5.10: Characterization of Bennu shape model using images from 0° latitude viewpoint (color of points indicate the errors)

5.4.1.2 Itokawa

A 200,000 facet reference shape model [41] is used to simulate the synthetic images of the asteroid Itokawa. These images capture Itokawa over its spin period of approximately 12.1 hrs [58] and thus each image is spaced approximately 9.1 min apart. Characterization of shape models generated using images from 0° and 45° latitudes are shown in Figs. 5.12 and 5.13 respectively. The equatorial images based shape model (Fig. 5.12a) contains 9,608 points with an overall RMS of 4.056 m (1.303% of the body radius). As the concavity along the equator (around $\pm 90^\circ$ longitude) is unobserved in the equatorial image set, the error in this region is higher (Fig. 5.12b). In comparison,

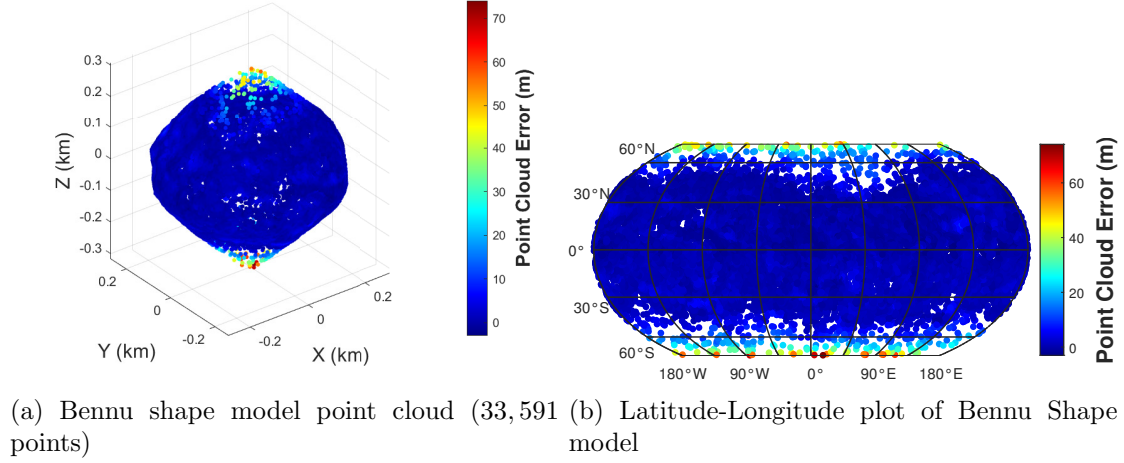


Figure 5.11: Characterization of Bennu shape model using images from 45° latitude viewpoint (color of points indicate the errors)

the shape model using 45° latitude images contains 23,054 points and shows improved performance in this region (Fig. 5.13b) with lower error and increased density of points. However, this shape model does not accurately characterize regions of Itokawa north of 45° and south of -45° latitudes due to unobservability of those regions. The overall RMS of this shape model is 4.672 m (1.50% of the body radius).

5.4.1.3 67P/C-G

A 200,000 facet reference shape model [21] is used to simulate the synthetic images of comet 67P/C-G. These images capture 67P/C-G over its spin period of approximately 12.4 hrs [86] and thus each image is spaced approximately 9.3 min apart. Characterization of shape models generated using images from 0° and 45° latitudes are shown in Figs. 5.14 and 5.15 respectively. The shape model derived from equatorial images (Fig. 5.15a) consists of 16,832 points with an overall RMS error of 53.99 m (2.038% of the body radius). Due to the unobservable concavity along the equator, the shape model error in this region is higher (Fig. 5.14b). In comparison, the shape model derived from 45° image set (consisting of 30,380 points) characterizes this region more accurately (Fig. 5.15b) but fails to accurately model the regions north of 45° latitude and south of -45°

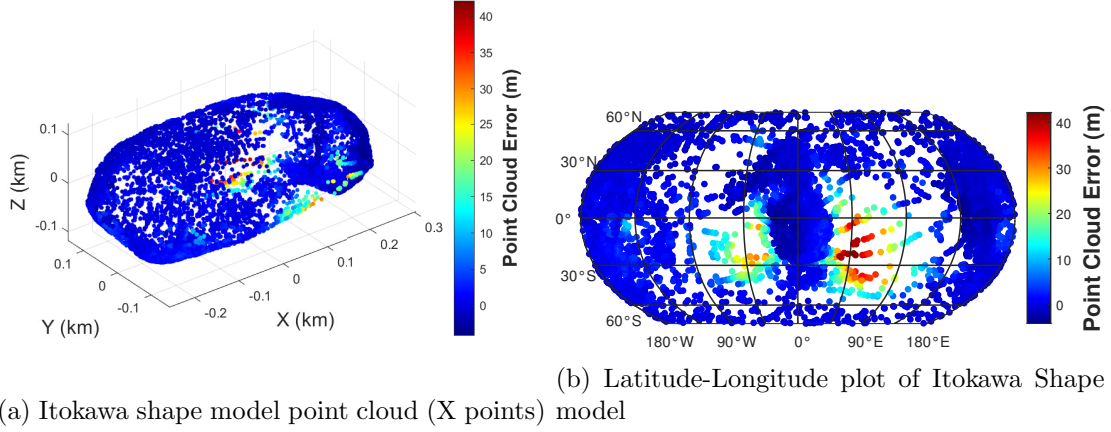


Figure 5.12: Characterization of Itokawa shape model using images from equatorial viewpoint (color of points indicate the errors)

latitude. The overall RMS error of this shape model is 54.27 m (2.049% of the body radius).

5.4.1.4 Eros

A 200,000 facet reference shape model [130] is used to simulate the synthetic images of asteroid Eros. These images capture Eros over its spin period of approximately 5.27 hrs [85] and thus each image is spaced approximately 3.95 min apart. Characterization of shape models generated using images from 0° and 45° latitudes are shown in Figs. 5.14 and 5.15 respectively. The shape model derived from equatorial images (Fig. 5.16a) consists of 8,216 points and has an overall RMS error of 341.6 m (1.938% of the body radius). The two major shape concavities along the broadside of Eros are not observable in these images (indicated by the absence of shape points in those regions). Thus, the error in these regions is higher (Fig. 5.16b). In comparison, the shape model derived from 45° (consisting of 20,219 points) captures these concavities more accurately. Thus, the overall RMS for this shape model is lower at 217.1 m (1.231% of the body radius). However, similar to the other bodies demonstrated earlier, the regions north of 45° latitude and south of -45° latitude are inaccurately captured in this shape model due to their unobservability in the images. The RMS error of the 45° latitude shape model (Fig. 5.17a) is notably smaller than that of the 0° latitude

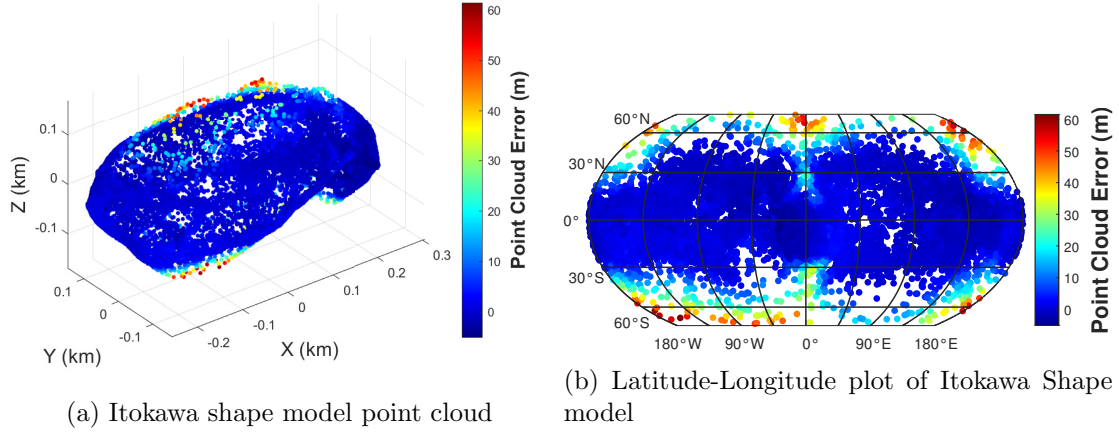


Figure 5.13: Characterization of Itokawa shape model using images from 45° latitude viewpoint (color of points indicate the errors)

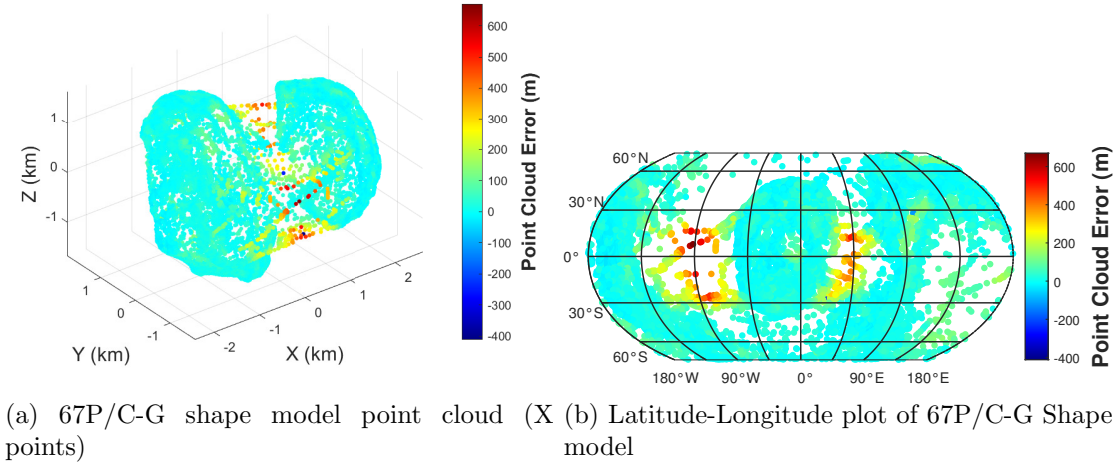


Figure 5.14: Characterization of 67P/C-G shape model using images from equatorial viewpoint (color of points indicate the errors)

shape model (Fig. 5.16a). This is because the increased observability of the equatorial concavities overrides the unobservability of other regions.

5.4.2 Shape Models using Images from Two Latitudes

In Section 5.4.1, a shape model was generated for the four small bodies using images at two different latitudes 0° and 45° . As observed in Section 5.4.1, any given latitude will have limitations

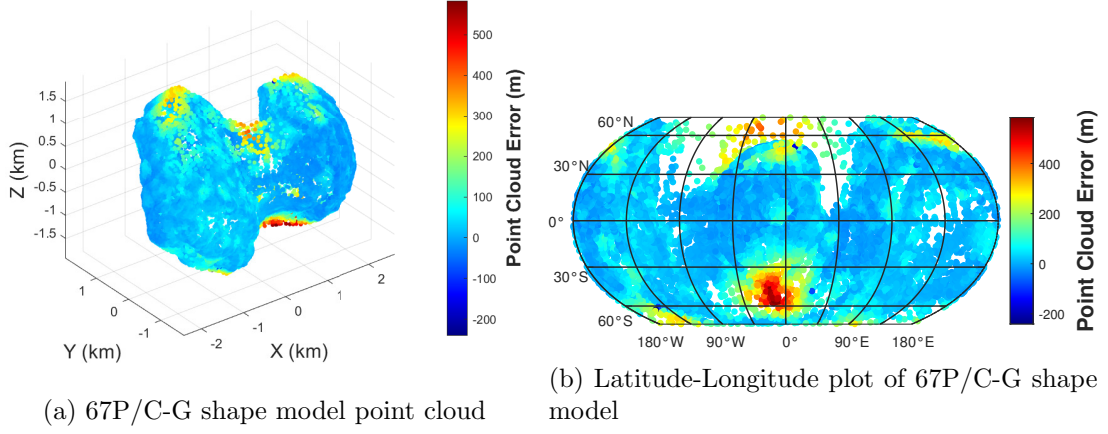


Figure 5.15: Characterization of 67P/C-G shape model using images from 45° latitude viewpoint (color of points indicate the errors)

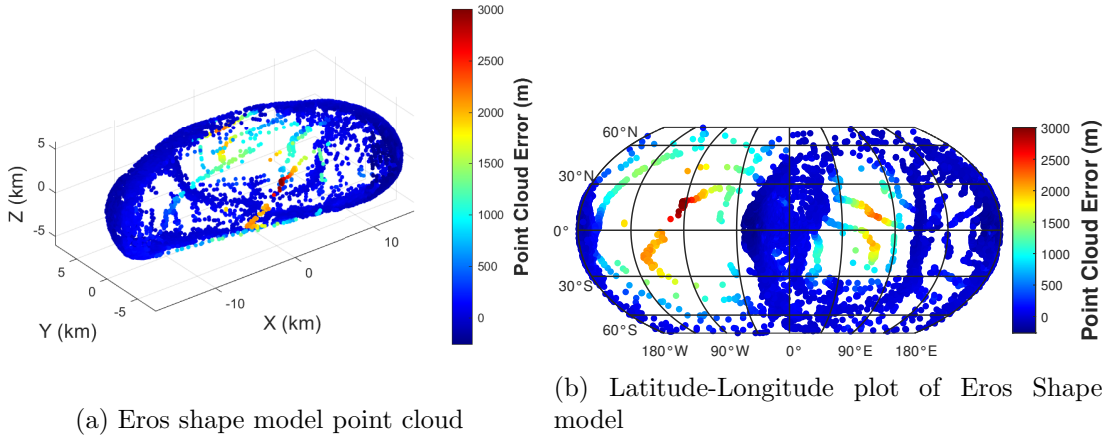


Figure 5.16: Characterization of Eros shape model using images from equatorial viewpoint (color of points indicate the errors). Absence of points indicate the limitation of the equatorial viewpoint.

accuracy in some regions and certain shape morphologies may be not well observed. In this section, shape models are derived using both sets of images to ascertain the accuracy improvements in combining different viewpoints. These shape models of Bennu, Itokawa, 67P/C-G, and Eros are shown in Fig. 5.18. A comparison of the 0° , 45° , and combined shape models are shown in Fig. 5.19 as a function of latitude.

Since asteroid Bennu is globally convex, the improvement in errors by combining the two

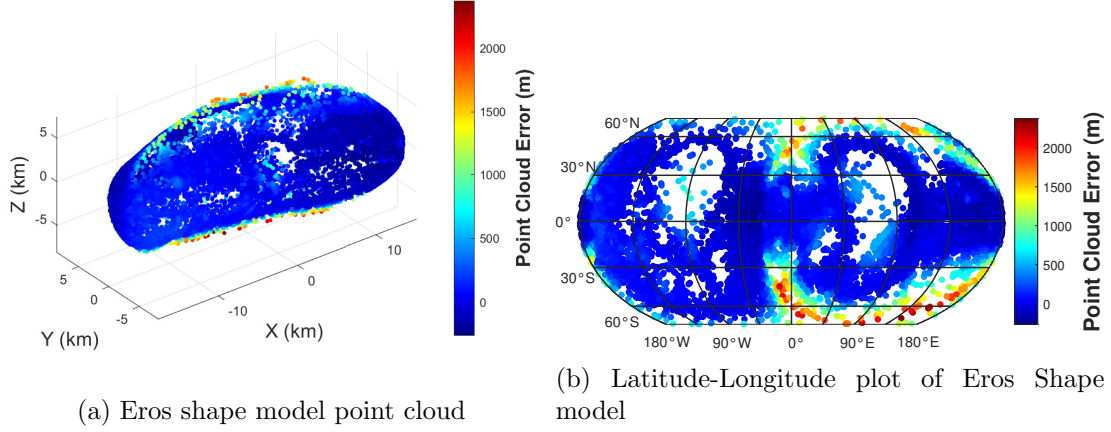


Figure 5.17: Characterization of Eros shape model using images from 45° latitude viewpoint (color of points indicate the errors)

views is marginal (Fig. 5.19a) with an overall shape model (with 47, 120 points) RMS error of 1.201 m (0.42% of the body radius). This RMS error is marginally better than the 1.334 m of the 0° shape model.

For Itokawa, 67P/C-G, and Eros, the equatorial concavities are captured more accurately (Figs. 5.18b to 5.18d). This can be seen in both the shape models (Figs. 5.18b to 5.18d) and in the latitude error plots (Figs. 5.19b to 5.19d). The combined shape models are able to accurately capture the concavities (like the 45° viewpoint) while showing good performance across latitudes (like the 0° viewpoint). The overall RMS of the Itokawa, 67P/C-G, and Eros are 1.5707 m, 25.552 m, and 115.47 m respectively (0.50%, 0.96%, and 0.65% of the body radii respectively). In comparison to the two other concave bodies, the equatorial concavity is not as accurately captured. This is due to the aforementioned limitation of any given viewing geometry. In the case of Eros, neither the 0° nor the 45° viewpoints are able to capture this concavity to a high degree of accuracy.

5.4.3 Sensitivity to Pole Errors

In this section, the algorithm's sensitivity to pole axis error is characterized for Bennu, Itokawa, CG/67-P, and Eros. It is shown in Chapter 3 and Ref. [65] the dominant error in the pole

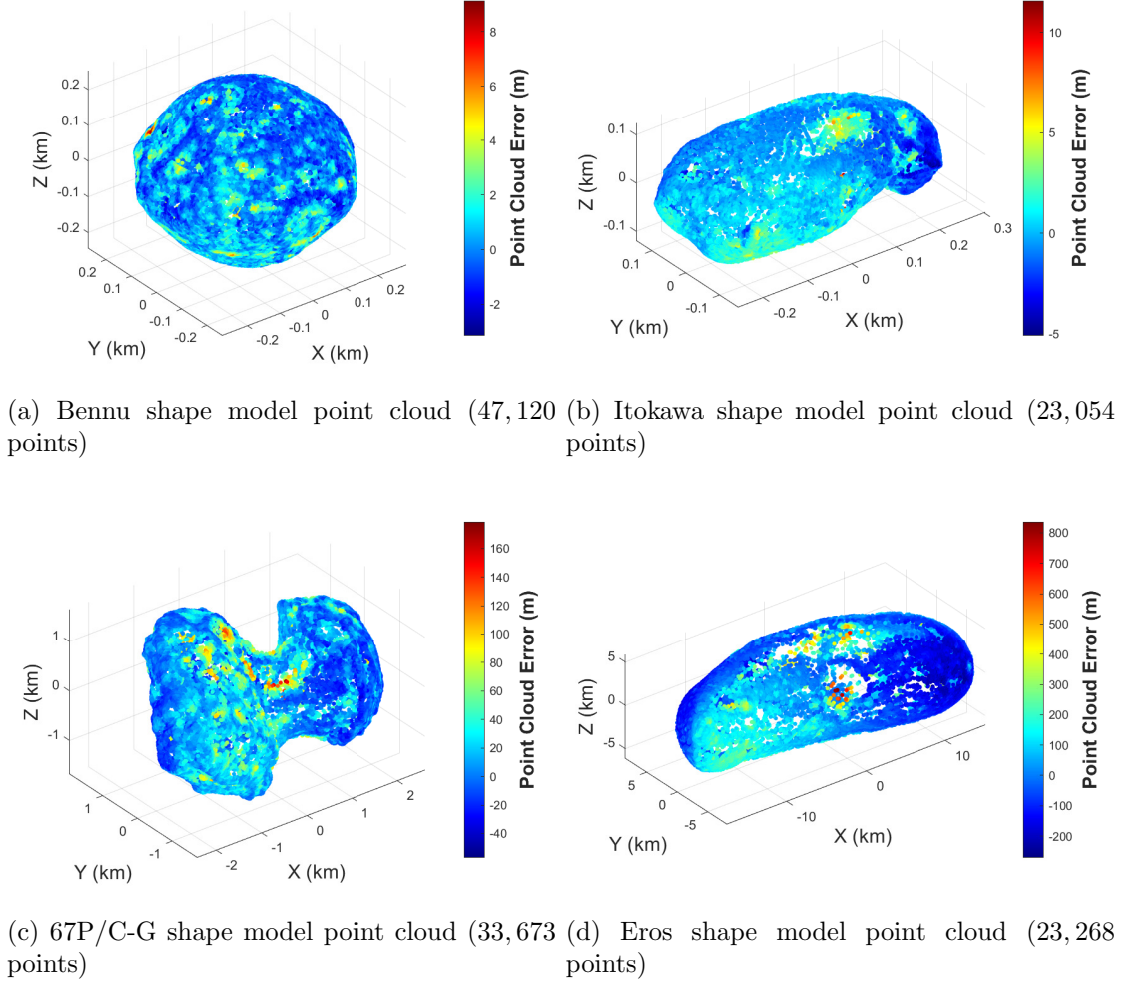


Figure 5.18: Shape models using images from 0° & 45° latitude viewpoints (color of points indicate the errors)

axis is in the latitude. Based on that, pole errors are induced in this work by adding a latitude error to the true pole to generate a perturbed pole orientation. With a true latitude of 0° , a pole error between 2° and 10° is added in increments of 2° . As the pole orientation determines the transformation between the camera and body frame axes ${}^{\mathcal{B}}\mathcal{R}$, errors in this parameter affect the location of the rays in the body frame (Eqs. (5.13) and (5.14)) and thus affect the shape modeling process. The statistics of the shape models with these pole errors are summarized in Table 5.2 and the shape models are shown in Fig. 5.20.

For Bennu, the overall accuracy in the 10° pole error case is high (less than 1% body radius

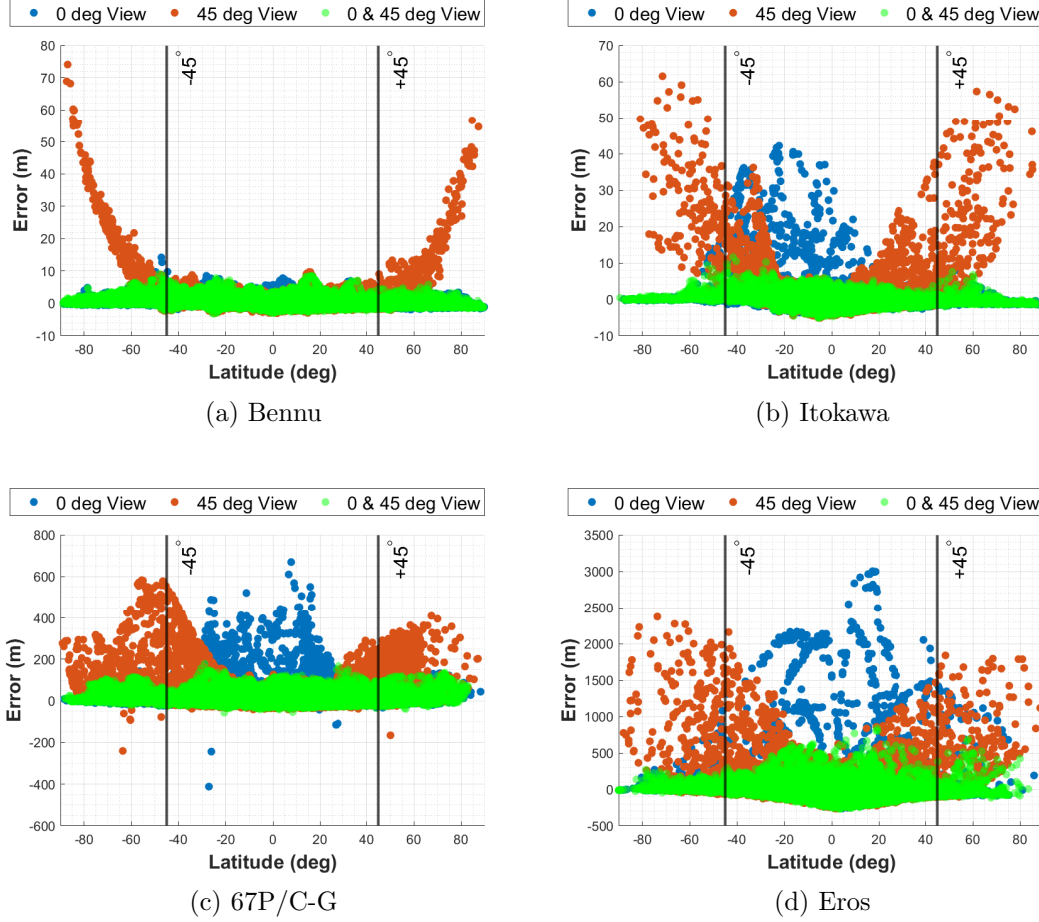
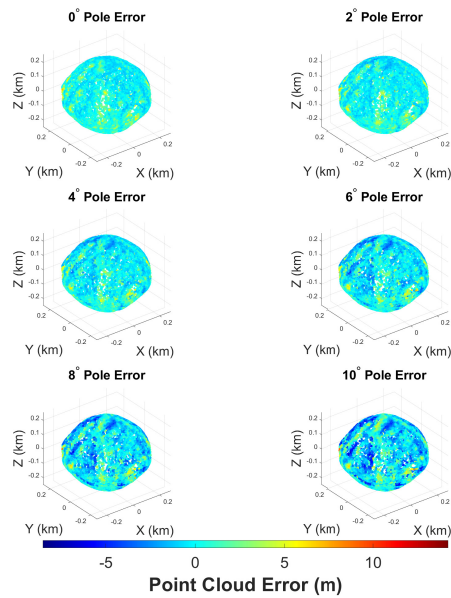


Figure 5.19: Comparison of performance across latitudes for 0° , 45° , and 0° & 45° cases

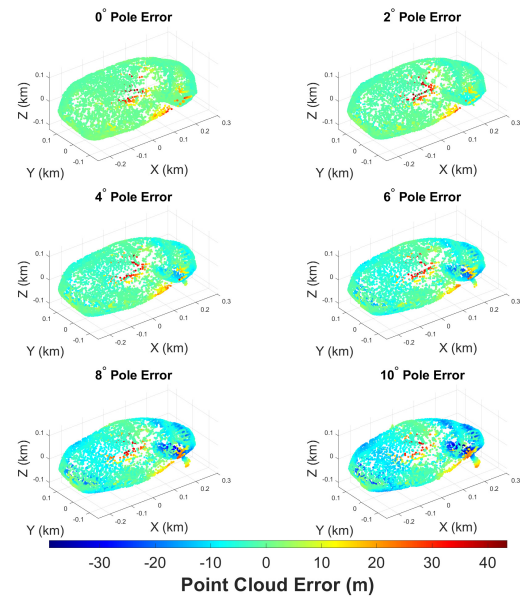
RMS error) due to its symmetric and globally convex shape. For Itokawa and Eros, the global shape is not as accurately captured at higher pole errors of 6° , 8° , and 10° (approximately 3 – 4% of the respective body radii). Also, a higher number of erroneous points are observed (especially at higher latitudes). Such erroneous points are also observed with 67P/C-G (Fig. 5.20c) on its left lobe. However, the global shape is captured better than Eros and Itokawa and similar to Benu. This is because of the elongated shapes of Eros and Itokawa. Points on the longer side of these bodies are further away from the axis of rotation (compared to the shorter side) and thus a pole error results in a higher degree of shape error in these regions due to a *lever-arm* effect. In contrast, since Benu and 67P/C-G are relatively less elongated, this effect is smaller.

Small Body	Pole Error ($^{\circ}$)	No. Points	RMS Error (m)	RMS Error (%)	Std Dev Error (m)	Std Dev Error (%)
Bennu [Fig. 5.20a]	2	16, 168	1.4023	0.48456	1.3813	0.47732
	4	15, 996	1.5162	0.52394	1.5153	0.52362
	6	15, 524	1.7270	0.59677	1.7221	0.59506
	8	15, 058	2.0106	0.69476	1.9840	0.68558
	10	14, 721	2.3454	0.81045	2.2919	0.79197
Itokawa [Fig. 5.20b]	2	8, 520	4.8770	1.5662	4.8140	1.5460
	4	7, 988	6.0303	1.9365	5.5476	1.7815
	6	7, 396	7.9592	2.5560	6.9303	2.2256
	8	7, 000	9.7913	3.1443	8.1407	2.6143
	10	6, 565	11.665	3.7460	9.2580	2.9731
67P/C-G [Fig. 5.20c]	2	13, 943	60.070	2.2678	58.115	2.1940
	4	12, 633	68.934	2.6024	68.368	2.5811
	6	11, 650	74.303	2.8051	74.306	2.8053
	8	10, 986	82.975	3.1325	82.475	3.1137
	10	10, 350	90.848	3.4298	88.879	3.3554
Eros [Fig. 5.20d]	2	6, 379	396.26	2.2476	396.18	2.2472
	4	5, 576	474.83	2.6933	464.23	2.6331
	6	5, 059	570.80	3.2376	542.19	3.0753
	8	4, 868	669.64	3.7982	623.87	3.5386
	10	4, 525	787.57	4.4671	723.91	4.1060

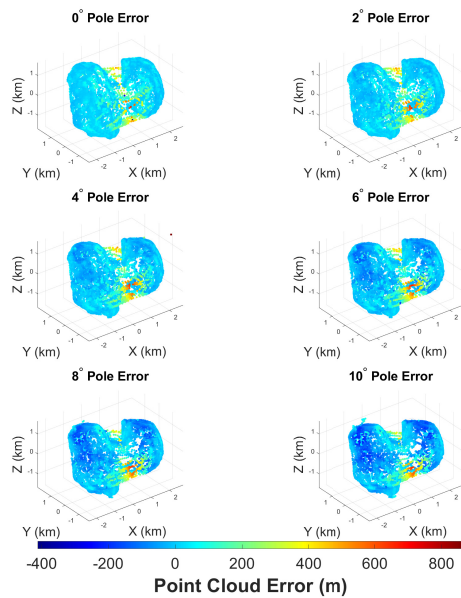
Table 5.2: Statistics of shape models with pole errors. RMS and Errors shown in units of meters and as a percentage of the body radii. References to the shape models are given in the leftmost column for the corresponding small body



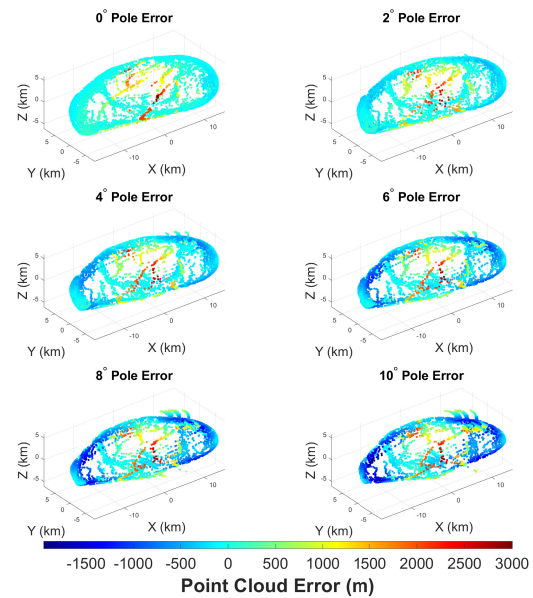
(a) Bennu



(b) Itokawa



(c) 67P/C-G



(d) Eros

Figure 5.20: Performance of Shape Models with Pole Error

5.4.4 Shape Model Derived from Hayabusa2 Images

The TIR camera [10] on-board the Hayabusa2 spacecraft took images of the asteroid Ryugu approximately 20 km from its center. A set of 64 images taken on October 20, 2018 [94] are used in this work. These images are taken over taken 7.84 hrs (1.0272x the spin period of 7.632 hrs) with images spaced approximately 7.5 min apart. Between two images, Ryugu rotates approximately 5.87° . The camera FOV is $16^\circ \times 12^\circ$ and with a resolution of 328×248 pixels [10]. In these images, the camera is viewing Ryugu approximately along its equatorial plane and thus all latitudes should be observable. Using these images, a shape model of the asteroid Ryugu is generated. The resultant point cloud consists of 1803 points and is characterized by comparing against a reference shape model [136] (Fig. 5.21). The overall RMS error of this model is 8.652 m (1.64% of the body radius). This demonstrates that the proposed algorithm can be used to produce an accurate shape model with real infrared images.

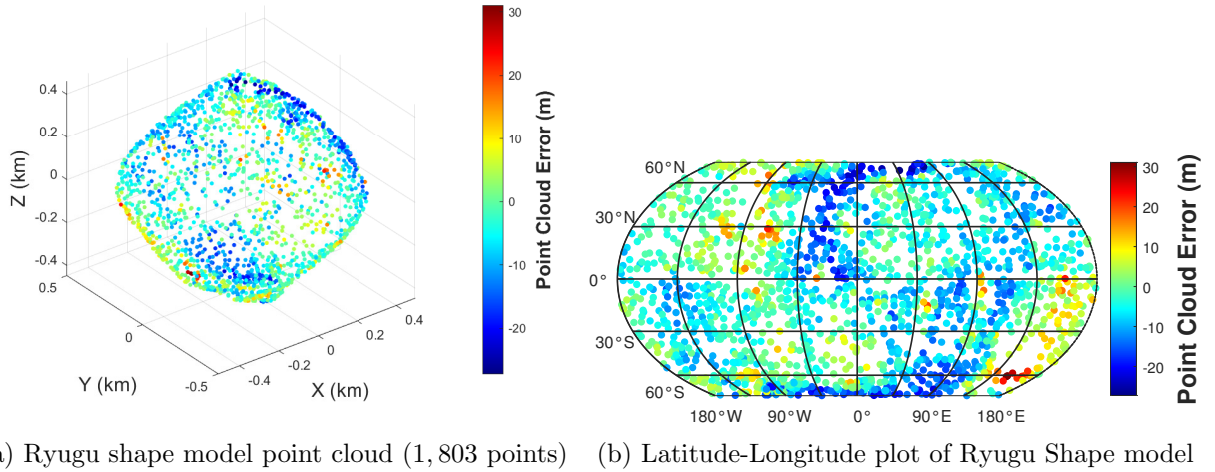


Figure 5.21: Characterization of Ryugu shape model using real images from Hayabusa2's TIR Camera (color of points indicate the errors)

5.5 Summary

In this chapter, a Shape from Silhouette (SfS) algorithm is presented to autonomously characterize the shape of a small body using a set of infrared images. In an image, the silhouette (visible extent) of the body is a robust source of shape information as it persists starting from when the body is resolved (*i.e.* spans more than a pixel). SfS algorithms can be applied as long as a resolved body is within the field-of-view (FOV) of a camera.

The pixels on the silhouette are extracted using an optimal sampling method (Section 5.3.1.3). This sampling method uses angles between a pixel with its neighboring pixels to determine whether additional shape information is obtained by the inclusion of the pixel in the shape extraction process. The reduction in the number of pixels greatly improves the computational efficiency while extracting the salient shape information from each image.

The shape extraction process (Section 5.3.2) casts a ray corresponding to each pixel in an image. From each neighboring pair of such rays, a plane segment is generated. Shape points are extracted by intersecting plane segments from different images and overlapping the resultant intersecting line segments (Section 5.3.4). The algorithm is applicable to convex shapes (*e.g.* Bennu, Ryugu) as well as concave shapes (*e.g.* Itokawa, 67P/C-G, Eros). Depending on the viewpoint of the images, certain concavities may not be observable. This unobservability can be mitigated by using images from several viewpoints. However, certain concave shape morphologies such as craters are not able to be estimated as they never appear on the silhouette.

The algorithm is applied to synthetic images of asteroids Bennu, Itokawa, and Eros as well as comet 67P/C-G. Two different approach latitudes are tested: 0° and 45° . The 0° images provide observability across latitudes with an RMS errors of 1.333 m, 4.056 m, 53.99 m, 341.6 m for Bennu, Itokawa, 67P/C-G, and Eros respectively (0.4609%, 1.0303%, 2.038%, 1.938% of the respective body radii). However, the equatorial concavities of Itokawa, 67P/C-G, and Eros are not well captured as seen in Figs. 5.12b, 5.14b and 5.16b. The 45° images produce shape models with RMS errors of 3.159 m, 4.672 m, 54.27 m, 217.1 m for Bennu, Itokawa, 67P/C-G, and Eros respectively (1.091%,

1.500%, 2.049%, 1.231% of the respective body radii). While the equatorial concavities are more accurately captured, the regions north of 45° and south of -45° are unobservable and hence not as well captured (Figs. 5.11b, 5.13b, 5.15b and 5.17b).

Combining images from both latitudes produces shape models with lower errors than either viewpoint (Figs. 5.18 and 5.19) with RMS errors of 1.201 m, 1.571 m, 25.55 m, 115.5 m for Bennu, Itokawa, 67P/C-G, and Eros respectively (0.4150%, 0.504%, 0.9647%, 0.6550% of the respective body radii). While the improvement for Bennu is marginal (due to its convex shape), the improvement in the other three bodies is noticeable.

The algorithm relies on knowledge of the pole axis to transform rays from each image to the body frame (Eqs. (5.13) and (5.14)). The sensitivity of the algorithm to errors in pole axis is characterized (Section 5.4.3). This analysis characterizes the increase in error as the pole error is increased. For Itokawa and Eros, higher pole errors ($> 6^\circ$) show significantly worse performance and the resultant shape models deviate significantly from the true shape of the bodies.

The algorithm is tested on real images of asteroid Ryugu taken by the TIR camera on-board JAXA's Hayabusa2 spacecraft. These images result in a shape model with an RMS error of 8.652 m (1.640% of the body radius). As Ryugu's shape is similar to that of Bennu, similar performance may be expected. However, the span of the Ryugu within the TIR images is smaller compared to the span of Bennu in the synthetic images. Due to this fewer pixels are extracted from each image of Ryugu which results in the worse performance.

While infrared images are more illumination-robust than optical images, the algorithm can be applied to the latter when the former are unavailable. The point cloud extraction steps (Section 5.3.2.5) remain the same regardless of the image type (optical or infrared) but the image processing steps may need to be modified for optical images depending on the solar phase angle. If the solar phase angle were zero, one can apply the algorithm without making any changes to the image processing steps. At non-zero phase angles, however, an additional step is necessary to extract the sun-lit section of the silhouette which is performed after the image binarization. A method to perform this extraction is given in Ref. [134]. The albedo of the body greatly affects the

apparent brightness of the body in optical images and the relative temperature difference between the background and the body is more distinct compared to the relative brightness difference between the background and the body (especially at low albedos). Thus, the background subtraction can be more reliably performed from infrared images regardless of the albedo properties of the body.

Chapter 6

Stereo Thermoclinometry

6.1 Introduction

Using the algorithm outlined in Chapter 5, an initial shape model of a small body is derived from the observed silhouettes in a set of infrared images. As noted in Chapter 5, SfS methods cannot accurately characterize concavities such as craters. As the temperature of the surface is well correlated with the shape, temperature measurements can be used to refine the shape of the body.

The algorithm presented in this chapter, named Stereo-thermoclinometry (STC), is designed to refine a small body shape using the measured temperatures of the surface from infrared images. STC first estimates the surface orientations (using a process called thermoclinometry) from a set of infrared images of a small body and then uses these estimates to solve for the vertices of the shape model (through an optimization problem). As direct measurements of the surface temperature are used, it is able to overcome limitations of SfS methods and characterize shape concavities. In addition to its robustness to illumination conditions, STC also alleviates some of the observational and computational constraints of Stereo-Photoclinometry [43] (SPC), the current state-of-the-art algorithm. As discussed in Ref. [42], SPC may not be able to produce accurate and global characterization of shadowed regions. This is especially true in flyby missions in which the observational time and geometry are limited. In contrast, the use of infrared images in STC allows for accurate characterization of surfaces even with limited observational times (Section 6.4.4). In addition, the photoclinometry steps of SPC require rendering of the images which is computationally

prohibitive. STC alleviates this computational burden by not necessitating the rerendering of images and thus removes the need for on-board rendering capabilities.

6.1.1 Outline

The rest of this chapter is structured as follows. The proposed method is detailed in Section 6.2. Uncertainty of the STC derived shape model is derived in Section 6.3. Results of the algorithm are illustrated in Section 6.4.

6.2 Stereo-Thermoclinometry (STC)

Starting with an initial shape model of a small body, the Stereo Thermoclinometry (STC) algorithm uses a thermo-physical model (TPM) to predict surface temperatures T_{pred} for each facet. These predictions are used in conjunction with measured temperatures from IR images T_{meas} to estimate the facet normal vectors $\hat{\mathbf{n}}_{est}$ of the shape model. This process is known as *thermoclinometry*. The shape model vertices corresponding to these estimated facet normals are solved using a constrained optimization scheme which results in a new shape model with updated vertices. The STC algorithm can be iterated until convergence which is achieved when the differences between predicted and measured surface temperatures are less than a specified threshold ϵ .

6.2.1 Thermo-Physical Model (TPM)

A Thermo-Physical Model (TPM) is used to model the temperatures of a given shape model. A general overview of TPMs is given in Section 2.2.2. In this chapter, the Astroshaper toolkit¹ [62] is used to model the temperatures of a given shape model.

6.2.2 Associating Pixel Temperatures to Facets

Each pixel in an infrared image of a small body provides a measurement of its surface temperature at a given time. Since the predicted temperatures are modeled on a shape model with

¹ The toolkit can be accessed here: <https://github.com/Astroshaper/AsteroidThermoPhysicalModels.jl>

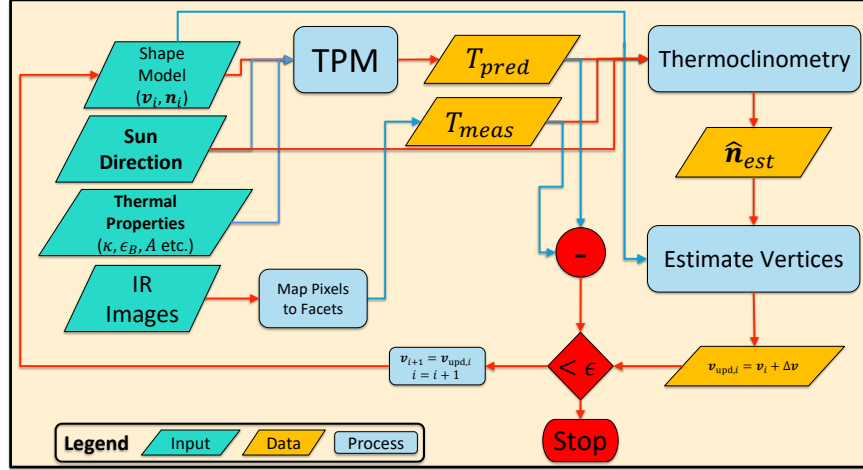


Figure 6.1: Flowchart for the STC Algorithm. T_{pred} and T_{meas} are the predicted and measured surface temperatures respectively. $\hat{\mathbf{n}}_{est}$ are the estimated surface orientations. $\mathbf{v}_i$ are the updated shape vertices

facets and vertices, an association process is necessary to map each pixel temperature to an associated facet on the shape model. This association process is similar to the rasterization process (Section 2.3.2) and is outlined in this section.

Let $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ be four pixels (illustrated in Fig. 6.2b) from an infrared image with associated temperatures T_1, T_2, T_3, T_4 respectively (Eq. (6.1)):

$$\begin{aligned}
 \mathbf{u}_1 &= [x_1, y_1] \\
 \mathbf{u}_2 &= [x_1, y_2] \\
 \mathbf{u}_3 &= [x_2, y_1] \\
 \mathbf{u}_4 &= [x_2, y_2]
 \end{aligned} \tag{6.1}$$

Let ${}_B\mathbf{f}_c$ be a facet center coordinate expressed in the body frame and shown in Fig. 6.2a. The homogenous coordinate projection of ${}_B\mathbf{f}_c$ is computed by projecting ${}_B\mathbf{f}_c$ into the camera using the camera position ${}_B\mathbf{r}$, the body to camera frame transformation matrix ${}_B^C\mathbf{R}$, and the camera

calibration matrix K as shown by Eqs. (6.2) and (6.3) where Z is the z-th component of ξ_f :

$$\xi_f = {}^c_B R({}_B \mathbf{f}_c - {}_B \mathbf{r}) \quad (6.2)$$

$$\bar{\mathbf{u}}_f = \frac{1}{Z} K \xi_f = \begin{bmatrix} \mathbf{u}_f \\ 1 \end{bmatrix} \quad (6.3)$$

The pixel coordinates of the projection (denoted $\mathbf{u}_f$) is obtained from the first two components of the homogenous vector $\bar{\mathbf{u}}_f$ from Eq. (6.3). The components of this pixel coordinate $\mathbf{u}_f$ are expressed in Eq. (6.4) and illustrated in Figs. 6.2a and 6.2b.

$$\mathbf{u}_f = [x_f, y_f] \quad (6.4)$$

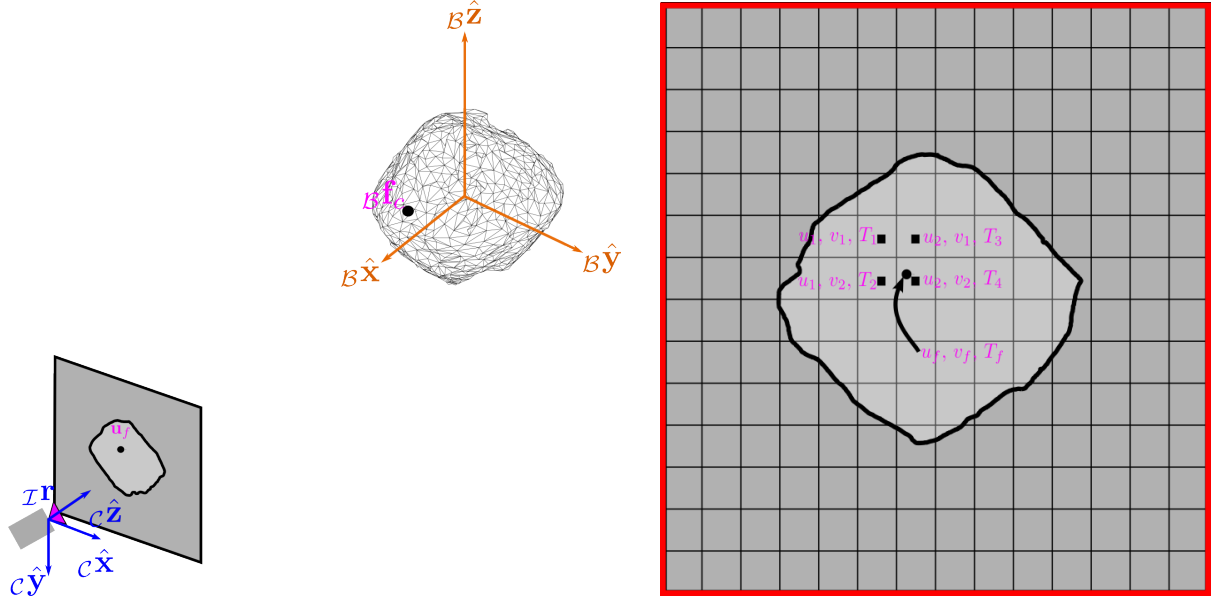
The measured temperature T_f of the facet (with pixel coordinate $\mathbf{u}_f$) is computed using the a weighted sum of temperatures T_1, T_2, T_3, T_4 Eq. (6.6) where the weighting vector $\mathbf{w}$ defined in Eq. (6.7) is a function of the coordinates of $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ and $\mathbf{u}_f$.

$$T_f = \mathbf{w} \cdot \mathbf{t} \quad (6.5)$$

$$\mathbf{t} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}^T \quad (6.6)$$

$$\mathbf{w} = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 y_2 & -y_2 & -x_2 & 1 \\ -x_2 y_1 & y_1 & x_2 & -1 \\ -x_1 y_2 & y_2 & x_1 & -1 \\ x_1 y_1 & -y_1 & -x_1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ x_f \\ y_f \\ x_f y_f \end{bmatrix} \quad (6.7)$$

This process is applied to all facets in view to obtain their measured temperatures. Additional precision in the interpolated temperatures may be gained by employing a higher fidelity interpolation schemes such as cubic or spline methods but these incur higher computational cost.



(a) 3D view of projection of a facet center $\mathcal{B}\mathbf{f}$ into the (b) 2D view of interpolating for a temperature of a facet T_f based on neighboring pixel temperatures

Figure 6.2: Illustration of mapping pixel temperatures to facets

6.2.3 Thermoclinometry

The thermoclinometry part of the STC algorithm consists of two steps: (1) estimate incidence angles, and (2) estimate facet normals which are outlined in Sections 6.2.3.1 and 6.2.3.2 respectively.

6.2.3.1 Estimate Incidence Angles

Recall that the surface boundary condition of a given facet (Eq. (2.7)) depends on the incidence angle α . Eq. (2.7) and can be rearranged to solve for the incidence angle as follows:

$$c\alpha = \cos \alpha = \frac{R_{AU}^2}{S_{\odot}(1-A)(1-s)} \left[\epsilon_B \sigma_{SB} T^4 - \kappa \left. \frac{dT}{dz} \right|_{surf} \right] \quad (6.8)$$

This equation can be expressed as a Taylor series around a reference temperature T_{ref} :

$$c\alpha = c\alpha_{ref} + \left. \frac{\partial c\alpha}{\partial T} \right|_{T=T_{ref}} (T - T_{ref}) \quad (6.9)$$

$$\frac{\partial c\alpha}{\partial T} = \frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} T^3 \quad (6.10)$$

Using Eq. (6.9), an estimate of the incidence angle is obtained using the measured and predicted temperatures (T_{meas} , T_{pred}):

$$c\alpha_{est} = c\alpha_{pred} + \Delta T \frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} T_{pred}^3 \quad (6.11)$$

$$\Delta T = T_{meas} - T_{pred} \quad (6.12)$$

Eq. (6.11) is used to estimate the incidence angles for all facets and indicates that when the measured temperature T_{meas} is higher than the predicted temperature T_{pred} (*i.e.* $\Delta T > 0$), the estimated incidence angle is smaller than predicted (*i.e.* $\alpha_{est} < \alpha_{pred}$). This implies that the facet is oriented closer towards the sun than predicted. Conversely, when the measured temperature is T_{meas} is lower than the predicted temperature T_{pred} (*i.e.* $\Delta T < 0$), Eq. (6.11) estimates the incidence angle to be larger than predicted (*i.e.* $\alpha_{est} > \alpha_{pred}$) implying that the facet is oriented further away from the sun direction than predicted.

6.2.3.2 Estimate Facet Normals

Recall that the incidence angle is the angle between the facet's normal vector $\hat{\mathbf{n}}$ and the sun direction $\hat{\mathbf{s}}$ (Fig. 2.4):

$$\cos \alpha = \hat{\mathbf{s}} \cdot \hat{\mathbf{n}} = \hat{\mathbf{s}}^T \hat{\mathbf{n}} \quad (6.13)$$

Using multiple estimates of the incidence angles (each at a different time), a linear system of equations can be written as follows (for a given facet):

$$\begin{bmatrix} \cos \alpha_1 \\ \cos \alpha_2 \\ \vdots \\ \cos \alpha_m \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{s}}_1^T \hat{\mathbf{n}} \\ \hat{\mathbf{s}}_2^T \hat{\mathbf{n}} \\ \vdots \\ \hat{\mathbf{s}}_m^T \hat{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{s}}_1^T \\ \hat{\mathbf{s}}_2^T \\ \vdots \\ \hat{\mathbf{s}}_m^T \end{bmatrix} \hat{\mathbf{n}} \quad (6.14)$$

An estimate of the facet normal can be obtained by solving Eq. (6.14) for $\hat{\mathbf{n}}$:

$$A = \begin{bmatrix} \hat{\mathbf{s}}_1^T \\ \hat{\mathbf{s}}_2^T \\ \vdots \\ \hat{\mathbf{s}}_m^T \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \cos \alpha_1 \\ \cos \alpha_2 \\ \vdots \\ \cos \alpha_m \end{bmatrix} \quad (6.15)$$

$$\hat{\mathbf{n}} = (A^T A)^{-1} A^T \mathbf{b} \quad (6.16)$$

Since the facet normal is a signed unit vector in $\mathbb{R}^3$, at least 3 incidence angle measurements are required to obtain a facet normal estimate (*i.e.* $m \geq 3$). In addition, the matrix A must be full rank in order for the matrix $A^T A$ to be invertible.

6.2.4 Deriving Corresponding Vertices

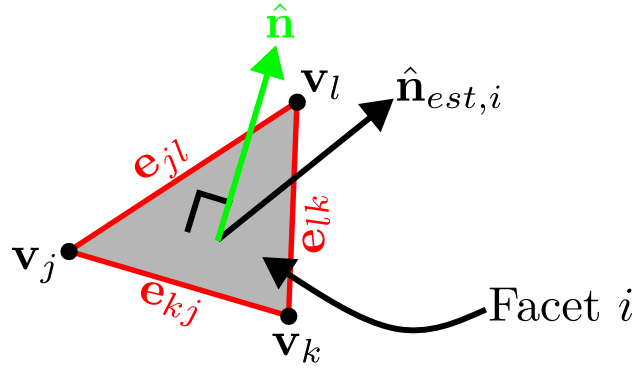


Figure 6.3: Vertices ($\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l$), Edges ($\mathbf{e}_{kj}, \mathbf{e}_{lk}, \mathbf{e}_{jl}$), True Normal ($\hat{\mathbf{n}}$), Estimated Normal ($\hat{\mathbf{n}}_{est,i}$) for a facet i

The thermoclinometry step of the STC algorithm (Section 6.2.3) is used to estimate facet normal vectors for all facets. An optimization problem is devised and outlined here to estimate the vertices based on the estimated facet normals. Consider the relationship between the three vertices of a triangular facet ($\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l$), and the facet normal $\hat{\mathbf{n}}$ defined by Eq. (6.17):

$$\hat{\mathbf{n}} = \frac{(\mathbf{v}_k - \mathbf{v}_j) \times (\mathbf{v}_l - \mathbf{v}_j)}{\|(\mathbf{v}_k - \mathbf{v}_j) \times (\mathbf{v}_l - \mathbf{v}_j)\|} \quad (6.17)$$

Let the three edges of a given facet (illustrated in Eq. (6.21)) be defined as follows:

$$\mathbf{e}_{kj} = \mathbf{v}_k - \mathbf{v}_j \quad (6.18)$$

$$\mathbf{e}_{lk} = \mathbf{v}_l - \mathbf{v}_k \quad (6.19)$$

$$\mathbf{e}_{jl} = \mathbf{v}_j - \mathbf{v}_l \quad (6.20)$$

Eq. (6.17) can be equivalently expressed in terms of any two of these three edges $\mathbf{e}_{kj}$, $\mathbf{e}_{jl}$, $\mathbf{e}_{lk}$ (Eq. (6.21)):

$$\hat{\mathbf{n}} = \frac{\mathbf{e}_{kj} \times \mathbf{e}_{jl}}{\|\mathbf{e}_{kj} \times \mathbf{e}_{jl}\|} \quad (6.21)$$

As these edges are in the plane of the facet, they are *orthogonal* to the facet normal $\hat{\mathbf{n}}$. Thus, the dot product of an edge and the true facet normal *must* be zero. Let a residual r_i be defined as the dot product of the edges of a facet i and its *estimated* normal $\hat{\mathbf{n}}_{est,i}$:

$$r_i = (\hat{\mathbf{n}}_{est,i} \cdot \mathbf{e}_{kj})^2 + (\hat{\mathbf{n}}_{est,i} \cdot \mathbf{e}_{jl})^2 + (\hat{\mathbf{n}}_{est,i} \cdot \mathbf{e}_{lk})^2 \quad (6.22)$$

This residual approaches zero when the vertices converge to the optimal values such that the corresponding edges are orthogonal to the estimated normal $\hat{\mathbf{n}}_{est,i}$. Based on this residual, the following optimization problem is used to solve for the corresponding vertices:

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \vdots \\ \mathbf{v}_N \end{bmatrix} \quad (6.23)$$

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} F(\mathbf{x}) \quad (6.24)$$

$$F(\mathbf{x}) = \sum_{i=1}^M r_i \quad (6.25)$$

The state $\mathbf{x}$ is the set of vertices to be optimized (Eq. (6.24)). The optimal state $\mathbf{x}^*$ is the one that minimizes the cost function $F(\mathbf{x})$ which is a sum of residuals over M facets. Other optimization cost

functions may be a *squared* sum of the residuals. In this case, that isn't necessary as each residual is positive by definition (Eq. (6.22)). The cost function $F(\mathbf{x})$ is minimized when the gradient is zero. This gradient is computed as follows:

$$\nabla F(\mathbf{x}) = \frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} = \sum_{i=1}^M \frac{\partial r_i}{\partial \mathbf{x}} \quad (6.26)$$

$$\frac{\partial r_i}{\partial \mathbf{x}} = 2 \hat{\mathbf{n}}_{est,i}^T \left[\mathbf{e}_{kj} \left(\hat{\mathbf{n}}_{est,i}^T \frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{x}} \right) + \mathbf{e}_{jl} \left(\hat{\mathbf{n}}_{est,i}^T \frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{x}} \right) + \mathbf{e}_{lk} \left(\hat{\mathbf{n}}_{est,i}^T \frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{x}} \right) \right] \quad (6.27)$$

$$\frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{v}_1} & \frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{v}_2} & \frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{v}_3} & \dots & \frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{v}_N} \end{bmatrix} \quad (6.28)$$

$$\frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{v}_1} & \frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{v}_2} & \frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{v}_3} & \dots & \frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{v}_N} \end{bmatrix} \quad (6.29)$$

$$\frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{v}_1} & \frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{v}_2} & \frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{v}_3} & \dots & \frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{v}_N} \end{bmatrix} \quad (6.30)$$

Based on the definition of the edges, the three jacobians (Eqs. (6.28) to (6.30)) are sparse matrices as summarized below:

$$\frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{v}_n} = \emptyset_{3 \times 3} \forall n \neq [k, j] \& n \in [1, N] \subset \mathbb{Z} \quad (6.31)$$

$$\frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{v}_k} = I_{3 \times 3} \quad (6.32)$$

$$\frac{\partial \mathbf{e}_{kj}}{\partial \mathbf{v}_j} = -I_{3 \times 3} \quad (6.33)$$

$$\frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{v}_n} = \emptyset_{3 \times 3} \forall n \neq [j, l] \& n \in [1, N] \subset \mathbb{Z} \quad (6.34)$$

$$\frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{v}_j} = I_{3 \times 3} \quad (6.35)$$

$$\frac{\partial \mathbf{e}_{jl}}{\partial \mathbf{v}_l} = -I_{3 \times 3} \quad (6.36)$$

$$\frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{v}_n} = \emptyset_{3 \times 3} \forall n \neq [l, k] \& n \in [1, N] \subset \mathbb{Z} \quad (6.37)$$

$$\frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{v}_l} = I_{3 \times 3} \quad (6.38)$$

$$\frac{\partial \mathbf{e}_{lk}}{\partial \mathbf{v}_k} = -I_{3 \times 3} \quad (6.39)$$

A Quasi-Newton optimization algorithm [90] is used to solve for the optimal state $\mathbf{x}^*$. Fig. 6.4 shows a flowchart of the algorithm. A search direction $\mathbf{p}_k$ at the current optimization step k is defined in Eq. (6.40) as a function of the hessian B_k and the gradient $\nabla F(\mathbf{x})$ respectively of the

cost function.

$$B_k \mathbf{p}_k = -\nabla F(\mathbf{x}) \quad (6.40)$$

Eq. (6.40) is solved to identify a search direction $\mathbf{p}_k$ given the current estimate of the hessian inverse H_k . Typically, an identity matrix I is set as the initial hessian inverse H_0 .

$$H_k \equiv B_k^{-1} \quad (6.41)$$

$$\mathbf{p}_k = -H_k \nabla F(\mathbf{x}) \quad (6.42)$$

A state update (defined in Eq. (6.44)) is computed based on this search direction $\mathbf{p}_k$ and a step size α_k . The state is updated by adding $\mathbf{s}_k$ to the current state $\mathbf{x}_k$ as shown in Eq. (6.43).

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \quad (6.43)$$

$$\mathbf{s}_k = \alpha_k \mathbf{p}_k \quad (6.44)$$

An appropriate step size α_k used in Eq. (6.44) is identified using an inexact line search algorithm [37]. The step size is chosen to satisfy the Wolfe Conditions [138, 139] stated in Eqs. (6.45) and (6.46). For Quasi-newton algorithms, the parameters c_1 and c_2 are set to be 10^{-4} and 0.9 respectively [90]. The first condition (Eq. (6.45)), also known as the Armijo rule [11], requires that the step size α_k sufficiently decreases the cost function F while the second condition (Eq. (6.46)) ensures that the step size is not too small.

$$F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \leq F(\mathbf{x}_k) + c_1 \alpha_k \nabla F(\mathbf{x}_k)^T \mathbf{p}_k \quad (6.45)$$

$$\nabla F(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k \geq c_2 \nabla F(\mathbf{x}_k)^T \mathbf{p}_k \quad (6.46)$$

The inverse hessian is updated using the BFGS formula given by Eq. (6.47) where $\mathbf{s}_k$ is the state update defined in Eq. (6.44) and $\mathbf{y}_k$ is the gradient approximation defined in Eq. (6.48)

$$H_{k+1} = H_k + \frac{(\mathbf{s}_k^T \mathbf{y}_k + \mathbf{y}_k^T H_k \mathbf{y}_k) (\mathbf{s}_k \mathbf{s}_k^T)}{(\mathbf{s}_k^T \mathbf{y}_k)^2} - \frac{H_k \mathbf{y}_k \mathbf{s}_k^T + \mathbf{s}_k \mathbf{y}_k^T H_k}{\mathbf{s}_k^T \mathbf{y}_k} \quad (6.47)$$

$$\mathbf{y}_k = \nabla F(\mathbf{x}_{k+1}) - \nabla F(\mathbf{x}_k) \quad (6.48)$$

A graphical illustration of the BFGS Quasi-Newton update step is given in Fig. 6.4 and is repeated until the optimizer achieves convergence.

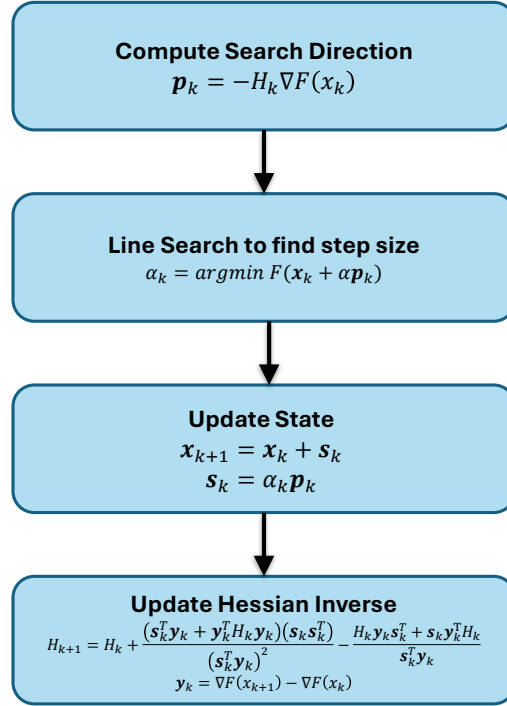


Figure 6.4: Quasi-Newton Update with a BFGS Update

6.2.5 STC Algorithm

The STC algorithm is summarized in Algorithm 4. The algorithm takes a set of images I and initial shape model facets F and vertices $\mathbf{x}_0$ as inputs. Line 2 applies the association process outlined in Section 6.2.2 to extract measured temperatures T_{meas} . Line 3 applies a thermo-physical model (TPM) to produce predicted temperatures T_{pred} . Lines 4 and 5 constitute the thermoclinometry step of the STC algorithm outlined in Section 6.2.3. Line 6 applies *one step* of the BFGS update outlined in Section 6.2.4. These steps are iterated until the stopping condition is met.

The three main steps of the STC algorithm (lines 3–6 of Algorithm 4) each improve key parameters of the algorithm: (1) Thermophysical Model (TPM) improves temperatures T_{pred} (2) thermoclinometry improves the surface orientations $\hat{\mathbf{n}}_{est}$, and (3) vertex optimization improves the shape vertices $\mathbf{x}$. By feeding the output of one into the other, a feedback loop is formed which ensures convergence of the algorithm especially under nonlinearity of the thermodynamics.

Algorithm 4 STC Algorithm**Input:** $I, F, \mathbf{x}_0$ **Output:** $\mathbf{x}_{sol}$

```

1: while  $|T_{meas} - T_{pred}| < \epsilon$  do
2:    $T_{meas} = \text{mapPixels}(I, F, \mathbf{x}_i)$  ▷ Section 6.2.2
3:    $T_{pred} = \text{thermoPhysicalModel}(F, \mathbf{x}_i)$  ▷ Section 2.2.2
4:    $c\alpha_{est} = c\alpha_{pred} + \Delta T \frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} T_{pred}^3$  ▷ Incidence angle estimate Eq. (6.11)
5:    $\hat{\mathbf{n}}_{est} = (A^T A)^{-1} A^T \mathbf{b}$  ▷ Update facet normal estimate Eq. (6.16)
6:    $\mathbf{x}_{i+1} = \text{quasiNewtonUpdate}(\mathbf{x}_i, \hat{\mathbf{n}}_{est})$  ▷ Apply Eq. (6.43)
7:    $i = i + 1$  ▷ Update iteration
8: end while

```

6.3 STC Uncertainty

In this section, the uncertainty in the shape vertices is derived from uncertainty in the temperature measurements. If the measurement errors are normally distributed with mean μ_T and variance σ_T^2 , the mean $\mu_{c\alpha}$ and variance $\sigma_{c\alpha}^2$ of $\cos \alpha$ is derived using the definition from Eq. (6.11).

$$\mu_{c\alpha} = \mathbb{E}[c\alpha] \quad (6.49)$$

$$\mu_{c\alpha} = \mathbb{E} \left[c\alpha_{pred} + (T_{meas} - T_{pred}) \frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} T_{pred}^3 \right] \quad (6.50)$$

$$\mu_{c\alpha} = c\alpha_{pred} + (\mu_T - T_{pred}) \frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} T_{pred}^3 \quad (6.51)$$

The variance in $\cos \alpha$ is derived using Eqs. (6.11) and (6.51):

$$\sigma_{c\alpha}^2 = \mathbb{E}[c\alpha^2] - \mu_{c\alpha}^2 \quad (6.52)$$

The square of the mean of $\cos \alpha$ is derived:

$$\mu_{c\alpha}^2 = c\alpha_{pred}^2 + (\mu_T - T_{pred})^2 \left(\frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} \right)^2 T_{pred}^6 \quad (6.53)$$

$$+ 2c\alpha_{pred}(\mu_T - T_{pred}) \frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} T_{pred}^3 \quad (6.54)$$

$$\mu_{c\alpha}^2 = c\alpha_{pred}^2 + (\mu_T^2 + T_{pred}^2 - 2\mu_T T_{pred}) \left(\frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} \right)^2 T_{pred}^6 \quad (6.55)$$

$$+ 2c\alpha_{pred}(\mu_T - T_{pred}) \frac{4R_{AU}^2 \epsilon_B \sigma_{SB}}{S_{\odot}(1-A)(1-s)} T_{pred}^3 \quad (6.56)$$

The expected value of the square of $\cos \alpha$ is derived:

$$\mathbb{E}[c\alpha^2] = \mathbb{E}\left[\left(c\alpha_{pred} + (T_{meas} - T_{pred})\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}T_{pred}^3\right)^2\right] \quad (6.57)$$

$$\mathbb{E}[c\alpha^2] = c\alpha_{pred}^2 + \mathbb{E}[T_{meas}^2 - 2T_{meas}T_{pred} + T_{pred}^2] \left(\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}\right)^2 T_{pred}^6 \quad (6.58)$$

$$+ 2c\alpha_{pred}(\mu_T - T_{pred})\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}T_{pred}^3 \quad (6.59)$$

$$\mathbb{E}[c\alpha^2] = c\alpha_{pred}^2 + \mathbb{E}[T_{meas}^2 - 2T_{meas}T_{pred} + T_{pred}^2] \left(\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}\right)^2 T_{pred}^6 \quad (6.60)$$

$$+ 2c\alpha_{pred}(\mu_T - T_{pred})\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}T_{pred}^3 \quad (6.61)$$

$$\mathbb{E}[c\alpha^2] = c\alpha_{pred}^2 + (\sigma_T^2 + \mu_T^2 + T_{pred}^2 - 2\mu_T T_{pred}) \left(\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}\right)^2 T_{pred}^6 \quad (6.62)$$

$$+ 2c\alpha_{pred}(\mu_T - T_{pred})\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}T_{pred}^3 \quad (6.63)$$

The variance is obtained by subtracting Eq. (6.56) from Eq. (6.63):

$$\sigma_{c\alpha}^2 = \sigma_T^2 \left(\frac{4R_{AU}^2\epsilon_B\sigma_{SB}}{S_{\odot}(1-A)(1-s)}\right)^2 T_{pred}^6 \quad (6.64)$$

Eqs. (6.51) and (6.64) are used to derive the mean and variance of the cosine of the incidence angle $\cos \alpha$ of a facet at a given time based on the expected measurement mean and variance. Let $\mathbf{b}$ be a set of these measurements across time (Eq. (6.65)). The expected value of $\mathbf{b}$ is expressed in Eq. (6.66).

$$\mathbf{b} = \begin{bmatrix} c\alpha_1 \\ c\alpha_2 \\ \vdots \\ c\alpha_m \end{bmatrix} \quad (6.65)$$

$$\mu_{\mathbf{b}} = \mathbb{E}[\mathbf{b}] = \begin{bmatrix} \mu_{c\alpha'_1} \\ \mu_{c\alpha'_2} \\ \vdots \\ \mu_{c\alpha'_m} \end{bmatrix} \quad (6.66)$$

Under the assumption that measurements are independent of each other, the covariance of $\mathbf{b}$ is expressed as a diagonal matrix (Eq. (6.67)). Components of $\mu_{\mathbf{b}}$ are computed using Eq. (6.51) and the diagonal components of $P_{\mathbf{b}}$ are computed using Eq. (6.64).

$$P_{\mathbf{b}} = \text{diag} \left(\begin{bmatrix} \sigma_{c\alpha_1}^2 & \sigma_{c\alpha_2}^2 & \dots & \sigma_{c\alpha_m}^2 \end{bmatrix} \right) \quad (6.67)$$

The expected value of the facet normal estimate is derived using Eqs. (6.16) and (6.66).

$$\mu_{\hat{\mathbf{n}}} = \mathbb{E}[\hat{\mathbf{n}}] = (A^T A)^{-1} A^T \mu_{\mathbf{b}} \quad (6.68)$$

The covariance of the facet normal estimate is similarly derived using Eqs. (6.16) and (6.67).

$$P_{\hat{\mathbf{n}}} \equiv \mathbb{E}[\hat{\mathbf{n}} \hat{\mathbf{n}}^T] - \mu_{\hat{\mathbf{n}}} \mu_{\hat{\mathbf{n}}}^T \quad (6.69)$$

$$= \mathbb{E}[(A^T A)^{-1} A^T \mathbf{b} \mathbf{b}^T A (A^T A)^{-T}] - (A^T A)^{-1} A^T \mu_{\mathbf{b}} \mu_{\mathbf{b}}^T A (A^T A)^{-T} \quad (6.70)$$

$$P_{\hat{\mathbf{n}}} = (A^T A)^{-1} A^T (\mathbb{E}[\mathbf{b} \mathbf{b}^T] - \mu_{\mathbf{b}} \mu_{\mathbf{b}}^T) A (A^T A)^{-T} \quad (6.71)$$

$$P_{\hat{\mathbf{n}}} = (A^T A)^{-1} A^T P_{\mathbf{b}} A (A^T A)^{-T} \quad (6.72)$$

As the facet normal estimate is constrained to be unit-norm ($\|\hat{\mathbf{n}}\| = 1$), the covariance matrix (Eq. (6.72)) has to be adjusted. This is accomplished by adding an correction factor suggested in Refs. [25, 143]. The resultant covariance matrix for a unit-norm facet normal estimate is given in Eq. (6.73).

$$P_{\hat{\mathbf{n}}} = (A^T A)^{-1} A^T P_{\mathbf{b}} A (A^T A)^{-T} + \left(1 - \frac{1}{\|\mu_{\hat{\mathbf{n}}}\|}\right)^2 \mu_{\hat{\mathbf{n}}} \mu_{\hat{\mathbf{n}}}^T \quad (6.73)$$

Eq. (6.73) is used to compute the covariance for each facet in a set of M facets. Let $\mathbf{y}$ be the set of facet normal estimates of M facets:

$$\mathbf{y} = \begin{bmatrix} \hat{\mathbf{n}}_{est,1} \\ \hat{\mathbf{n}}_{est,2} \\ \vdots \\ \hat{\mathbf{n}}_{est,M} \end{bmatrix} \quad (6.74)$$

The covariance in $\mathbf{y}$ is given in Eq. (6.75) assuming that the measurements are independent.

$$P_{\mathbf{y}} = \begin{bmatrix} P_{\hat{\mathbf{n}}_{est,1}} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & P_{\hat{\mathbf{n}}_{est,2}} & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & P_{\hat{\mathbf{n}}_{est,3}} & \dots & \mathbf{0}_{3 \times 3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \dots & \dots & P_{\hat{\mathbf{n}}_{est,M}} \end{bmatrix} \quad (6.75)$$

The sensitivity of $\mathbf{y}$ to the state $\mathbf{x}$ (the set of N vertices, Eq. (6.24)) is expressed as:

$$H = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \hat{\mathbf{n}}_{est,1}}{\partial \mathbf{v}_1} & \frac{\partial \hat{\mathbf{n}}_{est,1}}{\partial \mathbf{v}_2} & \dots & \frac{\partial \hat{\mathbf{n}}_{est,1}}{\partial \mathbf{v}_N} \\ \frac{\partial \hat{\mathbf{n}}_{est,2}}{\partial \mathbf{v}_1} & \frac{\partial \hat{\mathbf{n}}_{est,2}}{\partial \mathbf{v}_2} & \dots & \frac{\partial \hat{\mathbf{n}}_{est,2}}{\partial \mathbf{v}_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{\mathbf{n}}_{est,M}}{\partial \mathbf{v}_1} & \frac{\partial \hat{\mathbf{n}}_{est,M}}{\partial \mathbf{v}_2} & \dots & \frac{\partial \hat{\mathbf{n}}_{est,M}}{\partial \mathbf{v}_N} \end{bmatrix} \quad (6.76)$$

The components of Eq. (6.76) are zero if $\hat{\mathbf{n}}_{est,m}$ is *not* a function of $\mathbf{v}_n$ (Eq. (6.78)). The components of Eq. (6.76) are express in Eqs. (6.79) to (6.81) when $\hat{\mathbf{n}}_{est,m}$ *is* a function of $\mathbf{v}_n$ and are derived in Appendix C. The $[\cdot]_{\times}$ symbol indicates a skew symmetric matrix.

$$\hat{\mathbf{n}}_{est,m} = \frac{(\mathbf{v}_k - \mathbf{v}_j) \times (\mathbf{v}_l - \mathbf{v}_j)}{\|(\mathbf{v}_k - \mathbf{v}_j) \times (\mathbf{v}_l - \mathbf{v}_j)\|} = \frac{\mathbf{a}_m}{\|\mathbf{a}_m\|} \quad (6.77)$$

$$\frac{\partial \hat{\mathbf{n}}_{est,m}}{\partial \mathbf{v}_n} = \mathbf{0}_{3 \times 3} \forall n \neq [j, k, l] \text{ \& } n \in [1, \dots, N] \quad (6.78)$$

$$\frac{\partial \hat{\mathbf{n}}_{est,m}}{\partial \mathbf{v}_j} = \frac{[\mathbf{v}_l - \mathbf{v}_k]_{\times}}{\|\mathbf{a}_m\|} \left(I_{3 \times 3} - \frac{\mathbf{a}_m \mathbf{a}_m^T}{\mathbf{a}_m^T \mathbf{a}_m} \right) \quad (6.79)$$

$$\frac{\partial \hat{\mathbf{n}}_{est,m}}{\partial \mathbf{v}_k} = \frac{[\mathbf{v}_j - \mathbf{v}_l]_{\times}}{\|\mathbf{a}_m\|} \left(I_{3 \times 3} - \frac{\mathbf{a}_m \mathbf{a}_m^T}{\mathbf{a}_m^T \mathbf{a}_m} \right) \quad (6.80)$$

$$\frac{\partial \hat{\mathbf{n}}_{est,m}}{\partial \mathbf{v}_l} = \frac{[\mathbf{v}_k - \mathbf{v}_j]_{\times}}{\|\mathbf{a}_m\|} \left(I_{3 \times 3} - \frac{\mathbf{a}_m \mathbf{a}_m^T}{\mathbf{a}_m^T \mathbf{a}_m} \right) \quad (6.81)$$

The matrix H (Eq. (6.76)) relates the sensitivity between the state $\mathbf{x}$ and the measurements $\mathbf{y}$. Given the covariance in the measurement $P_{\mathbf{y}}$, the covariance in the state is derived using H :

$$P_{\mathbf{x}} = (H^T P_{\mathbf{y}}^{-1} H)^{-1} \quad (6.82)$$

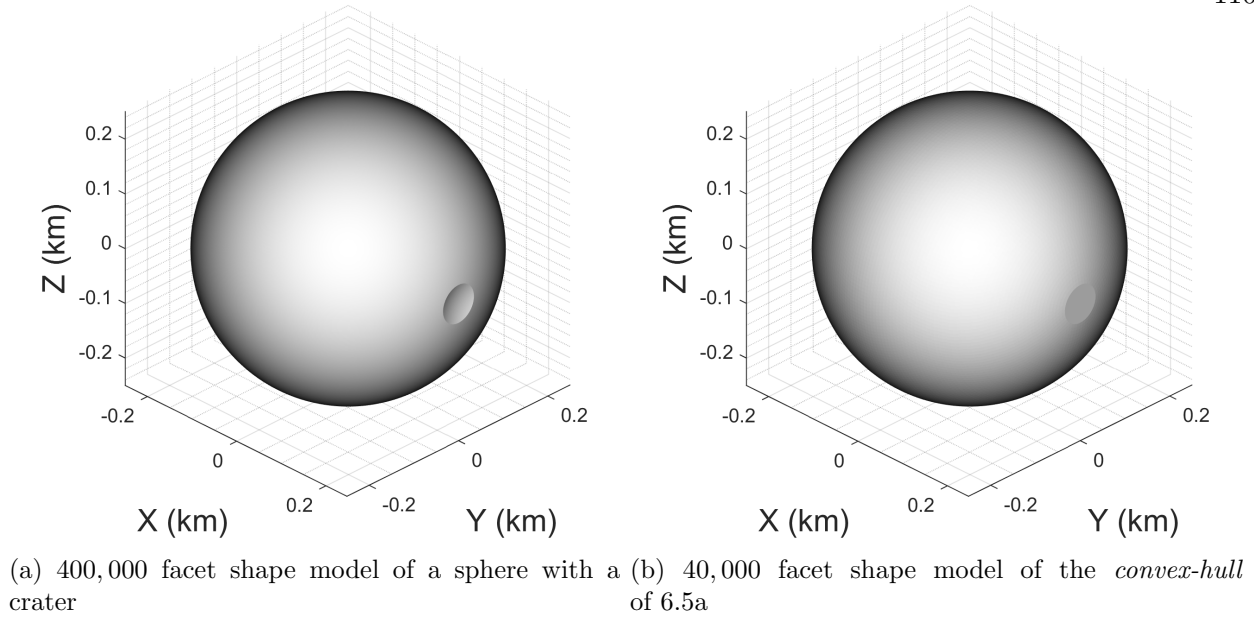


Figure 6.5: True and initial shape models of a spherical body with a crater

6.4 Results

In this section, the STC algorithm is applied to refine a shape model of a small body with a crater. The initial shape model is a convex hull of the true shape model which is expected from an SfS algorithm such as the one from Chapter 5. These two shape models are shown in Fig. 6.5. The crater has a depth of 2.3 m and a diameter of 33.6 m. The error in the initial shape model is given in Fig. 6.6 and it is used in all cases presented in this chapter. The Astroshaper TPM [62] outlined in Section 2.2.2 is applied to the true shape model (Fig. 6.5a) to generate true temperatures of each facet T_{true} over one spin period of this body. Material and thermal properties of the asteroid Ryugu, shown in Table 6.1, are used in this simulation. To simulate sensor noise, random gaussian error with zero mean and variance σ_T^2 of 1 K² is added to the true temperatures. Using these temperatures, images are generated using the rendering scheme outlined in Section 2.3.2 using the camera and observational parameters shown in Table 6.2. In this scenario, the camera is in a body fixed hovering position 20 km from the center of mass (CoM) the small body along the 0° latitude and 0° longitude. The sensitivity of the algorithm to different viewpoints is characterized

in Section 6.4.1.

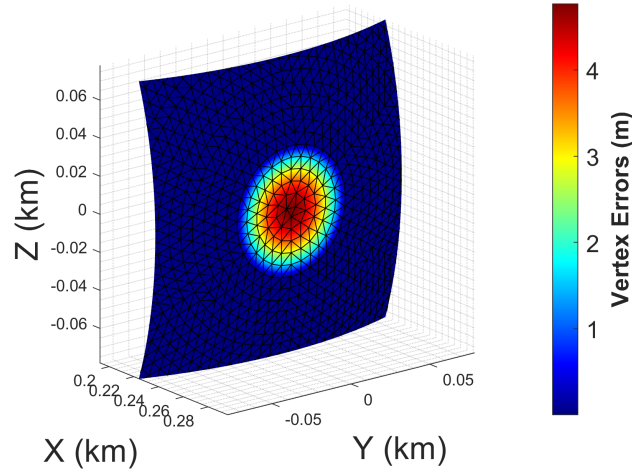


Figure 6.6: Initial shape model error

Table 6.1: Thermal parameters for sphere with crater shape model (using Ryugu parameters)

Parameter	Name	Value
κ	thermal conductivity	$0.1 \text{ W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$
ρ	material density	$1270 \text{ kg}\cdot\text{m}^{-3}$
C_p	heat capacity	$600 \text{ kg}\cdot\text{m}^2\cdot\text{s}^{-2}\cdot\text{K}^{-1}$
A	bond albedo	0.04
ϵ_B	thermal emissivity	1.0
σ_{SB}	Stefan-Boltzmann constant	$5.670374419 \times 10^{-8} \text{ W}\cdot\text{m}^{-2}\cdot\text{K}^{-4}$
P_{rot}	spin period	7.632 hrs
$S_{\odot}$	solar flux	$1366 \text{ W}\cdot\text{m}^{-2}$
R_{AU}	distance from the sun	$\sim 0.988 \text{ AU}$
γ	axial tilt	$\sim -7.16^\circ$

Table 6.2: Camera and observation parameters

Parameter	Value
FOV (field of view)	5°
Image Resolution	500×500 pixels
Camera Latitude (in body frame)	0°
Camera Longitude (in body frame)	0°
Camera Distance (from center of mass (CoM))	20 km
Number of Images (N)	121

The STC algorithm outlined in Section 6.2 and summarized in Algorithm 4 is applied to this

shape model. The Astroshaper TPM [62] is used to predict the temperatures using the initial shape model T_{pred} . The thermoclinometry and optimization steps of the STC algorithm are applied to update the vertices of the shape model (corresponding to the crater). This process is repeated for 60 iterations to characterize the convergence behavior of the algorithm. This parameter is a heuristic which was observed to be large enough for the algorithm to converge and is used throughout Section 6.4 to allow for comparative assessment of convergence across different cases and sensitivity analyses.

Fig. 6.7a shows the final shape model with the colors indicating the shape model error. Fig. 6.7b shows the errors of the crater vertices at each iteration of the STC algorithm. This latter figure illustrates the convergence of the algorithm over the iterations. The RMS error of the initial shape model is 3.04 m where as the RMS error of the final shape model (after 60 iterations) is 0.220 m which represents an improvement of approximately 93 %. The convergence also suggests that the algorithm achieves a steady state after approximately 30 iterations with minimal performance gains thereafter. The vertex errors shown on the final shape model indicate a difference in performance between the “northern” and “southern” vertices. This occurs due to the sensitivity of the algorithm to the axial tilt (γ from Fig. 2.1) and is characterized in Section 6.4.2.

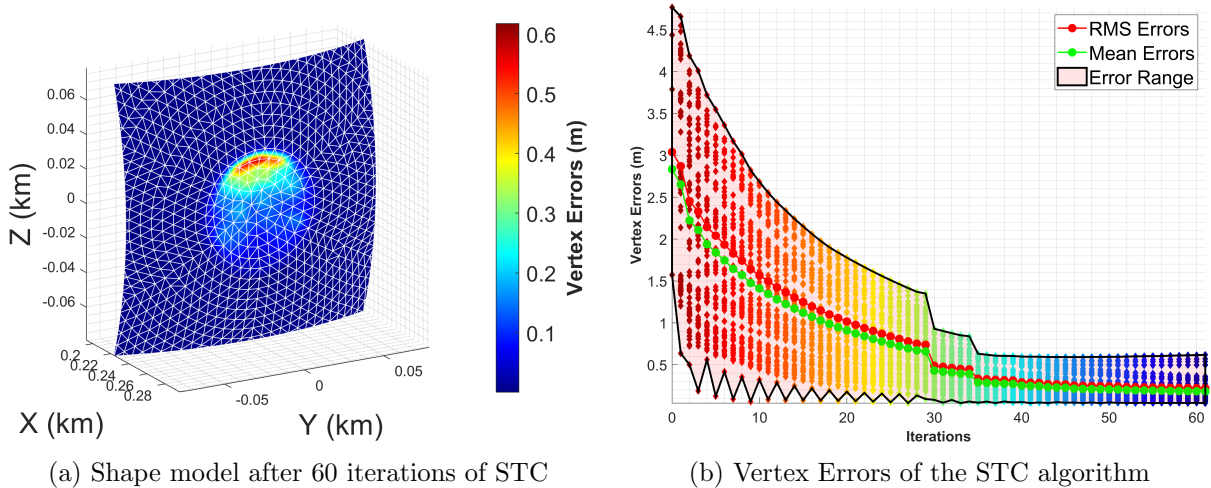


Figure 6.7: Performance of the STC algorithm for a sphere shape model with a crater

6.4.1 Observational Viewpoint Sensitivity Analysis

In this section, the sensitivity of the algorithm to the camera latitude and longitude is characterized. To associate the pixel measurements to the respective facet on the shape model, the process outlined in Section 6.2.2 is applied. Depending on the viewpoint, this association process can introduce errors. To illustrate this, consider the two plots shown in Fig. 6.8 in which the initial and true shape model of lower resolution (~ 40000 facets) are projected into the image plane. Fig. 6.8a shows an image from 0° latitude and 0° longitude. In this case, the true and initial shape models align and thus no association errors occur. Fig. 6.8b shows an image from 40° latitude and 50° longitude which shows that the projections of the true and initial shape models do not align (for the facets corresponding to the crater).

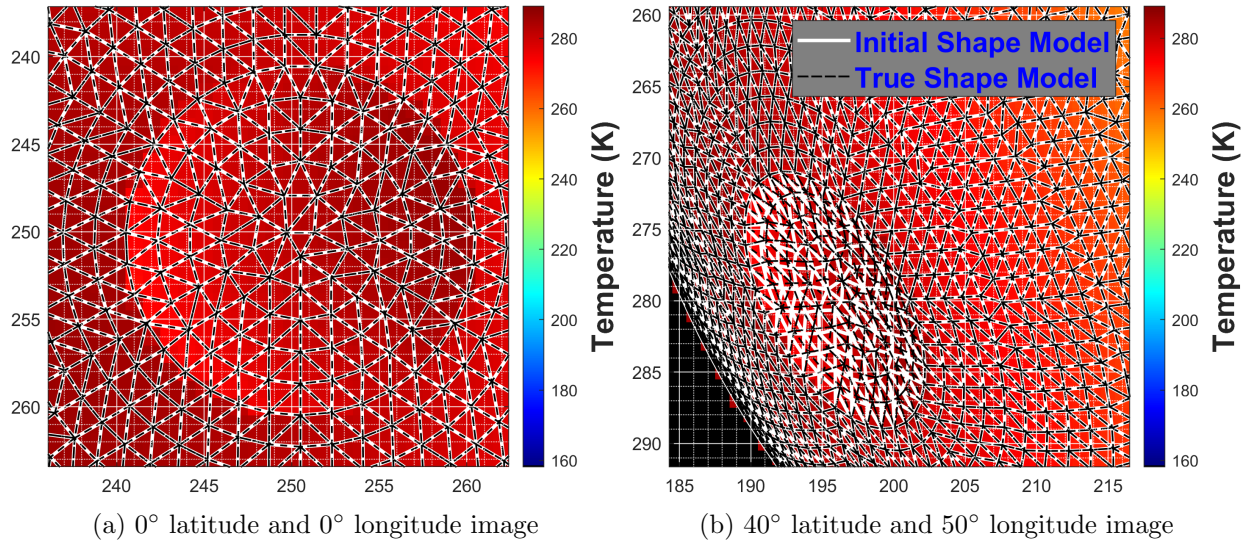


Figure 6.8: Two images at different latitudes and longitudes showing the projection error. The true shape model is shown in black and the initial shape model is shown in white

This misalignment occurs due to the projection issues and causes incorrect associations in the mapping process. These incorrect associations result in errors in measured temperatures for the facets of the crater. While the size and depth of the crater has an effect on this projection error, the observational geometry also has a significant effect as shown in Fig. 6.8. The sensitivity of the STC algorithm to the projection error and the observational geometry is characterized by

the illustration shown in Fig. 6.9. Each case shown in Fig. 6.9 corresponds to images taken from a different latitude and longitude. Each subplot shows the errors at a different iteration of the algorithm. Since the projection errors are lower latitudes and longitudes close to zero, these cases converge faster while the others converge slower. The final errors shown in Fig. 6.9f indicate that the observational geometries farther away from 0° latitude and 0° longitude have larger errors due to the aforementioned projection error. However, the algorithm is still able to reduce the errors significantly in all cases. Even in the worse case, -60° latitude and -50° longitude, the errors are reduced by 66 % compared to the initial guess. As the algorithm converges, the projection issues are reduced since the shape model approaches the true shape. Due to these incremental improvements to the shape model after each iteration of the STC algorithm, the algorithm is able to overcome the initial projection issues in most instances.

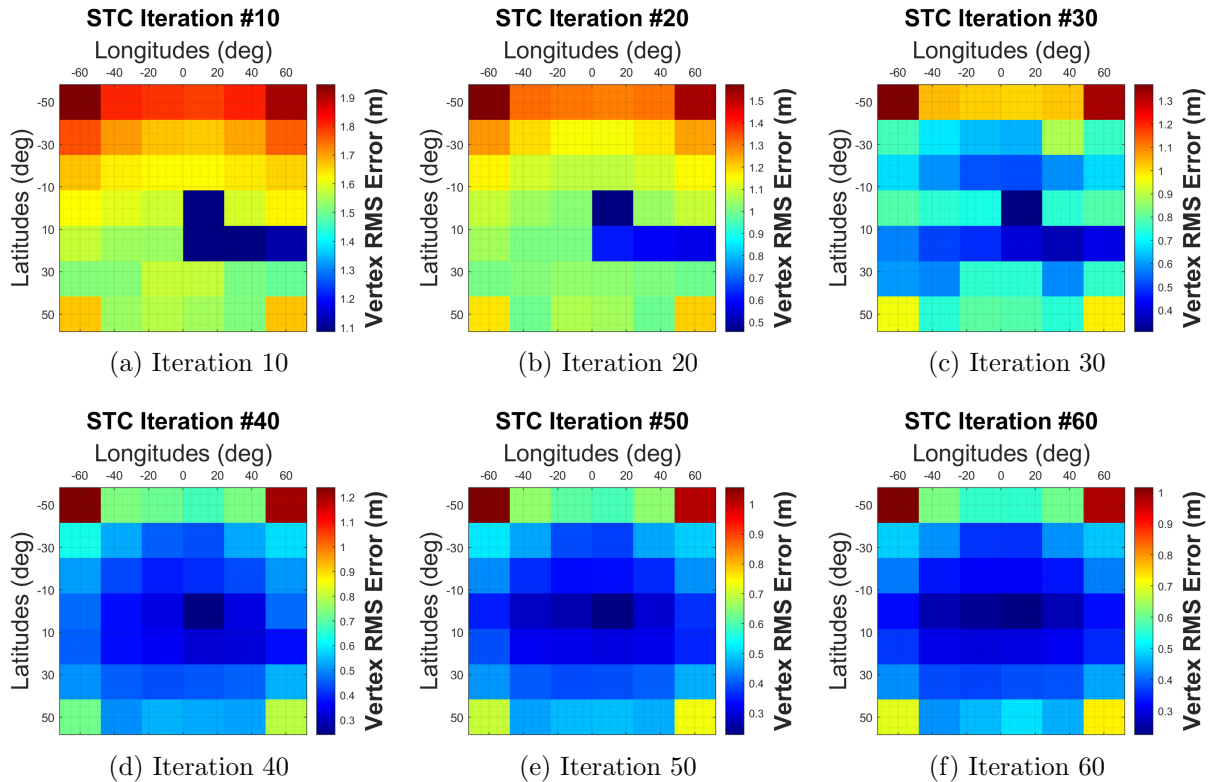


Figure 6.9: RMS errors to illustrate viewpoint sensitivity at different iterations of STC (note that the colormap are different across the sub-figures)

6.4.2 Axial Tilt Sensitivity

The shape model shown in Fig. 6.7a shows a difference in performance between the “northern” and “southern” sections of the crater with the former section performing worse in comparison to the latter. This occurs due to the sensitivity of the thermoclinometry step of the STC algorithm to the axial tilt γ which is the angle between the orbital plane and the equatorial plane (illustrated in Fig. 2.1). To understand this sensitivity, several axial tilts between -9° and $+9^\circ$ are tested. Apart from the axial tilt, the thermal and observational parameters given in Tables 6.1 and 6.2 respectively are used in this analysis. Fig. 6.10 shows the RMS error across the axial tilts. This indicates that axial tilts close to 0° (those that align with the mean plane of the crater) have worse error.

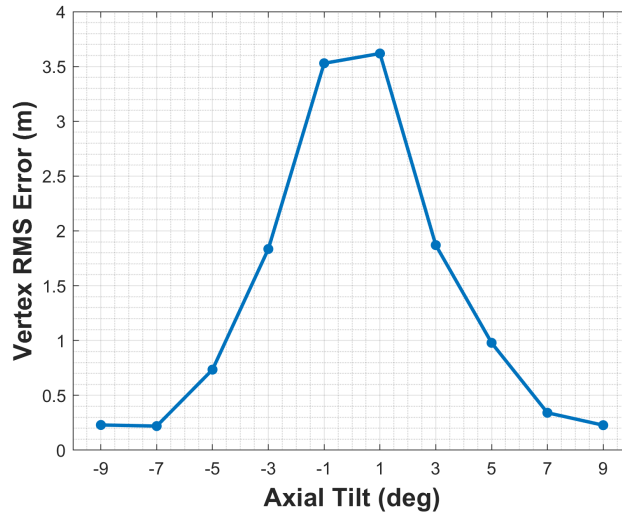


Figure 6.10: Shape model RMS across axial tilts after 60 iterations of the STC algorithm

To further characterize the axial tilt sensitivity, consider the shape models shown in Fig. 6.11. These shape models show that for negative axial tilts, the southern side of the crater has lower error in comparison to the northern sides. Conversely, for positive axial tilts, the northern side performs better. This sensitivity study shows that the algorithm performs better when the axial tilt is *not* close to the mean latitude of the section of the surface that is being estimated (note that for this shape model, the mean latitude of the crater is 0°). There is no notable difference

in shape model performance across longitudes of the crater. With more negative axial tilts, the northern facets of the crater have a smaller variation in the incidence angle over time compared to the southern facets. This prohibits the algorithm from accurately characterizing these facet normal orientations which leads to poorer shape performance. Conversely, with more positive axial tilts, the southern facets experience this phenomenon.

6.4.3 Image Frequency Analysis

In the case shown in Figs. 6.7a and 6.7b, 121 images across one spin period of the asteroid were used. This equates to approximately a 3.7 min spacing between images. In this section, the performance of the algorithm is characterized with a lower sampling rate (*i.e.* fewer images spaced across the spin period) to understand the behavior of the algorithm. Apart from the number of images N , the thermal and observational parameters given in Tables 6.1 and 6.2 respectively are used. Note that the thermophysical model (TPM) used the prediction step of the STC algorithm still simulates temperatures at the baseline rate (resulting in 121 observations) to ensure numerical integration accuracy. Fig. 6.12 shows the RMS error of the shape model after 60 iterations across different sampling rates. The difference in errors between any of the cases shown in Fig. 6.12 is less than 1 cm and therefore it is considered to be statistically insignificant. This analysis shows that the algorithm requires very few images across the spin period in order to converge on the true shape. Measurement coverage of the day/night cycle (across the spin period) is more important than the measurement frequency. This is due to the facet normal estimation step of the thermoclinometry process (Section 6.2.3). The incidence angles for a facet over a spin period is a cosine curve according the definition from Eq. (6.13). Accurate characterization of facet normal requires accurate characterization of this curve. The latter requires sampling across the entire curve to capture the peak and trough of this sinusoid. A measurement gap (*e.g.* only using the day or night time data) will miss either the peak or the trough and hence will result in inaccurate facet normals. This is especially true in cases where the initial shape model errors are high. To further illustrate phenomenon, consider the true, initial, and estimated incidence angles across the spin

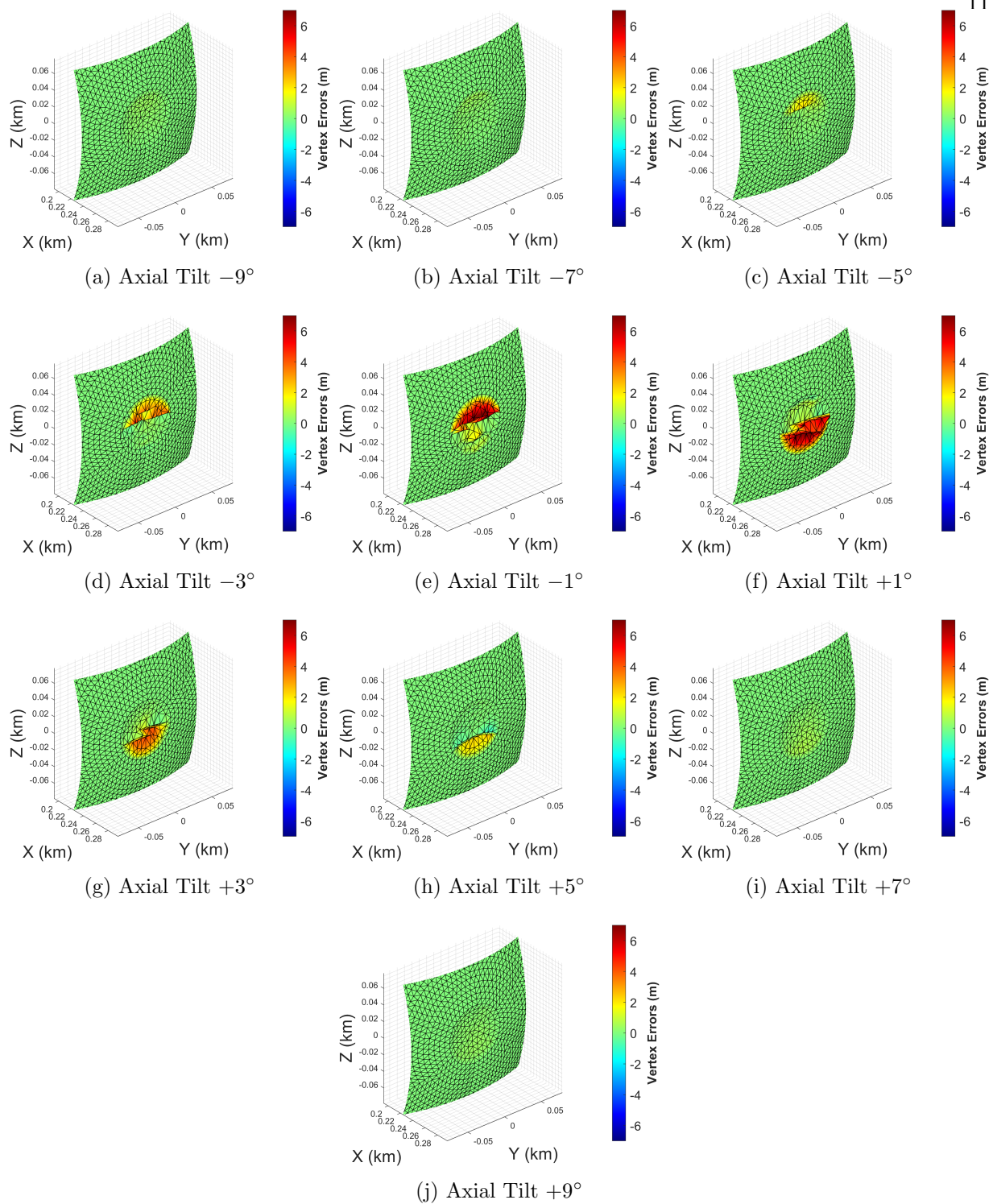


Figure 6.11: Shape models across different axial tilts between -9° and $+9^\circ$ after 60 iterations of the STC algorithm

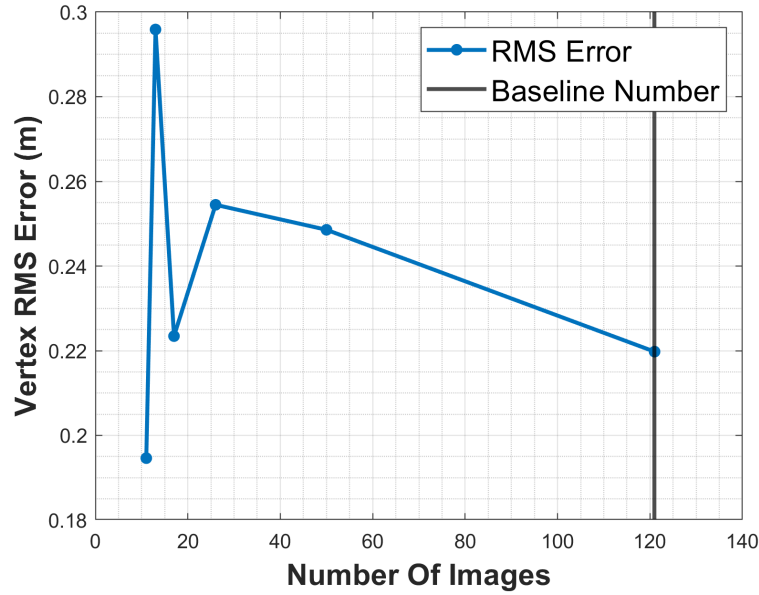


Figure 6.12: RMS errors of shape model after 60 iterations of STC across at different sampling rates

period shown in Fig. 6.13 for facet 140 of the shape model. Also shown in Fig. 6.13 are the predicted and measured temperatures obtained from the TPM and IR images respectively. If nighttime data is ignored (between time indices of 25 and 85) the trough of the sinusoid (around time index of 50) would be missed. This causes Eq. (6.16) to produce an inaccurate facet normal estimate as it is unable to accurately characterize the cosine curve. Thus, to ensure robust estimation of the facet normals, both night and day time data should be used.

6.4.4 STC with Flyby Images

In this section, images of a small body taken during a flyby trajectory are used to generate a shape model using the STC algorithm. A flyby trajectory is scenario in which the spacecraft approaches the body along the terminator plane (*i.e.* the plane orthogonal to the sun direction). The trajectory can be segmented into three sections (shown in Fig. 6.14): (1) inbound section along the terminator starting and ending at distances of 20 km and 10 km respectively (shown in red), (2) performs a flyby around the body, (3) leave along the terminator plane starting and ending at distances of 10 km and 20 respectively (shown in blue). Images taken during the inbound and

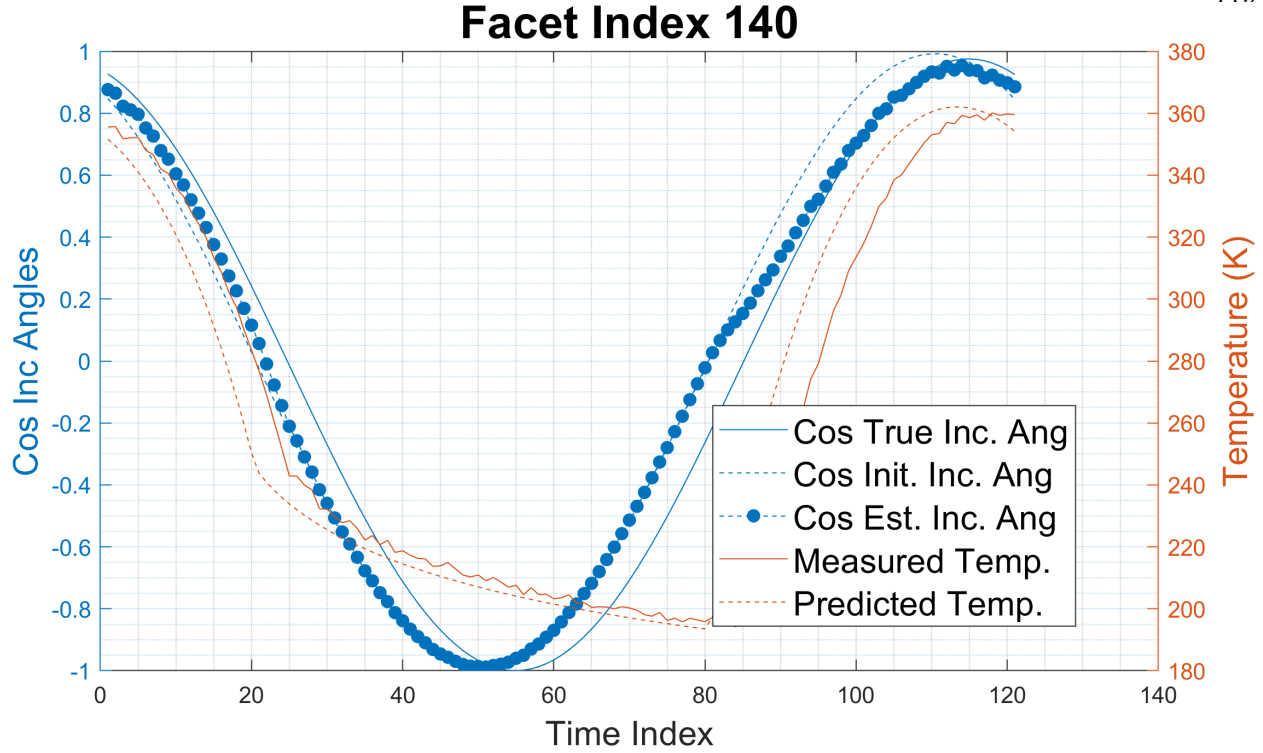


Figure 6.13: Initial, true, and estimated incidence angles for facet 140 of the shape model. Also shown are the predicted and measured temperatures obtained from the TPM and IR images respectively

outbound segments are used to estimate the body's shape. Each of these segments spans one half of the spin period of the body and are phased such that the combined set provides coverage over most of the spin period. The STC algorithm is applied to the flyby infrared images given the initial shape model shown in Fig. 6.5b. This initial shape model is implicitly assumed to be obtained using an SfS algorithm but is explicitly assumed to be known here. The true and initial shapes of the body shown in Fig. 6.5 are used here and parameters from Table 6.1 are used in modeling the temperatures.

Results of the STC algorithm are shown in Fig. 6.15. The final shape model RMS error is 0.224 m which is lower than that of the baseline case (shown in Fig. 6.7). This occurs as fewer nighttime images are used in the flyby scenario in comparison to the baseline scenario. As these images are less sensitive, their exclusion helps the algorithm achieve better performance. This suggests that more daytime images than nighttime images should be used to achieve better shape

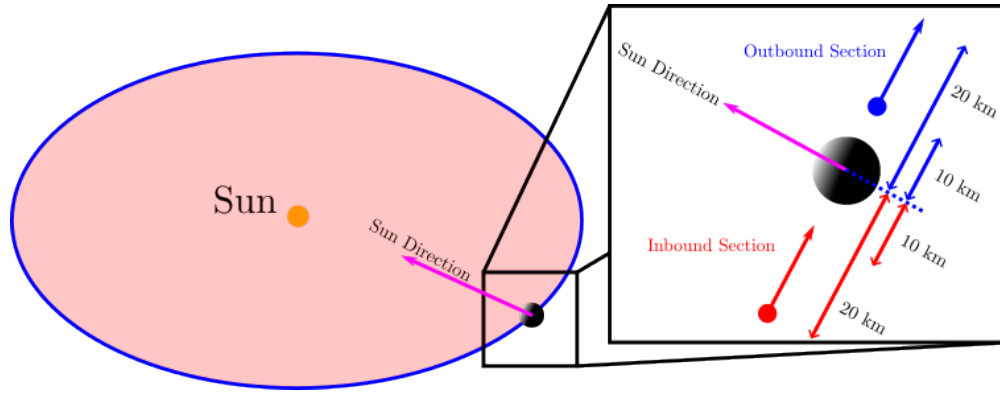


Figure 6.14: Geometry of the Flyby Scenario. Inbound section shown in red, outbound section shown in blue

estimate. Moreover, this performance suggests that the algorithm is capable of estimating shapes in flyby missions which is generally not possible for SPC since optimal viewing geometry of the latter may not be reached.

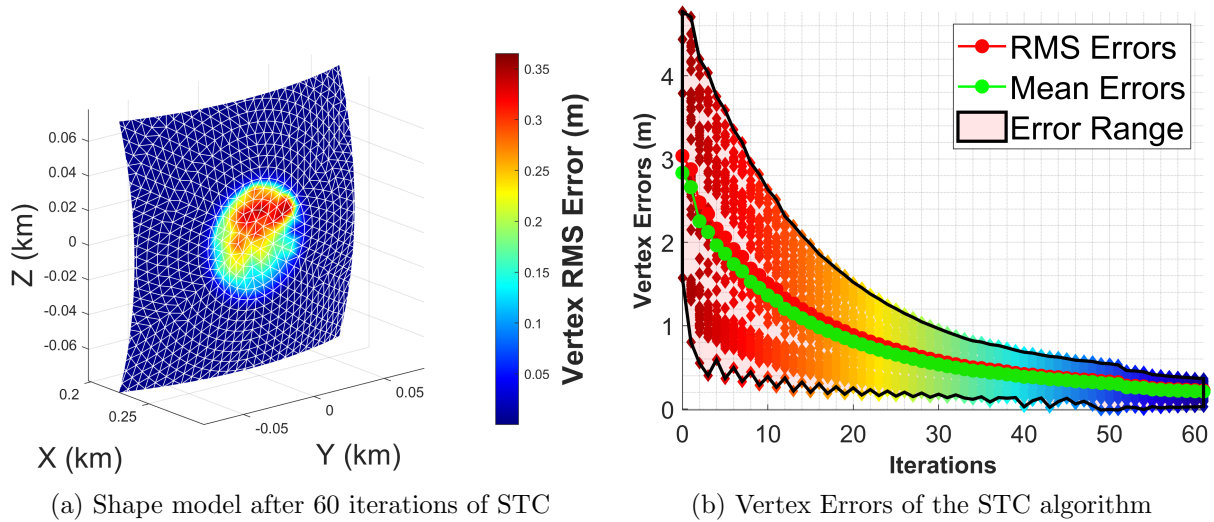


Figure 6.15: Performance of the STC algorithm for a sphere shape model with a crater in a Flyby scenario illustrated in Fig. 6.14

6.5 STC Uncertainty Monte Carlo

In this section, the STC analytical uncertainty derived in Section 6.3 is verified against Monte Carlo runs. A true shape model of the sphere with crater shown in Fig. 6.5a is used here but at a lower resolution of 10,000 facets. The TPM outlined in Section 2.2.2 is used to simulate the temperatures of this shape model and used as the predicted temperatures T_{pred} . Measured temperatures T_{meas} are generated by adding zero mean gaussian noise with variance σ_T^2 of $1K^2$ to T_{pred} as follows:

$$T_{meas} = T_{pred} + \epsilon_T \quad (6.83)$$

$$\epsilon_T \sim \mathcal{N}(0, \sigma_T^2) \quad (6.84)$$

For each Monte Carlo run, different noise ϵ_T (Eq. (6.84)) is used which generates different measured temperatures T_{meas} . For each such run, one iteration of the STC algorithm is applied to update the shape. As the predicted shape is the truth, doing so illustrates the sensitivity of the algorithm to the measurement error ϵ_T . Monte Carlo results of the incidence angles, facet normals and vertex positions are compared against analytical derived covariances derived from Section 6.3. The Monte Carlo results and the analytical uncertainty of the incidence angle errors (for one facet) are shown in Fig. 6.16. The estimates appear to be within the expected uncertainty bounds. As the uncertainty in the incidence angle is a function of the predicted temperature (as shown in Eq. (6.64)), it decreases during the nighttime as the temperature decreases. The Monte Carlo results in Fig. 6.16 validates this phenomenon.

In each of the Monte Carlo runs, the estimated incidence angles are used to estimate the facet normal. Fig. 6.17 shows a plot of the facet normal errors along with the analytical uncertainty (Eq. (6.72)) in cartesian space (results are shown for facet 1). The Monte Carlo runs are within the analytically derived uncertainty bounds. This validates the analytical uncertainty given by Fig. 6.17.

Using the quasi-newton optimization scheme, the vertices are updated based on the estimated normals for each Monte Carlo run. Fig. 6.18 shows the estimated vertices (for vertex 1) along with

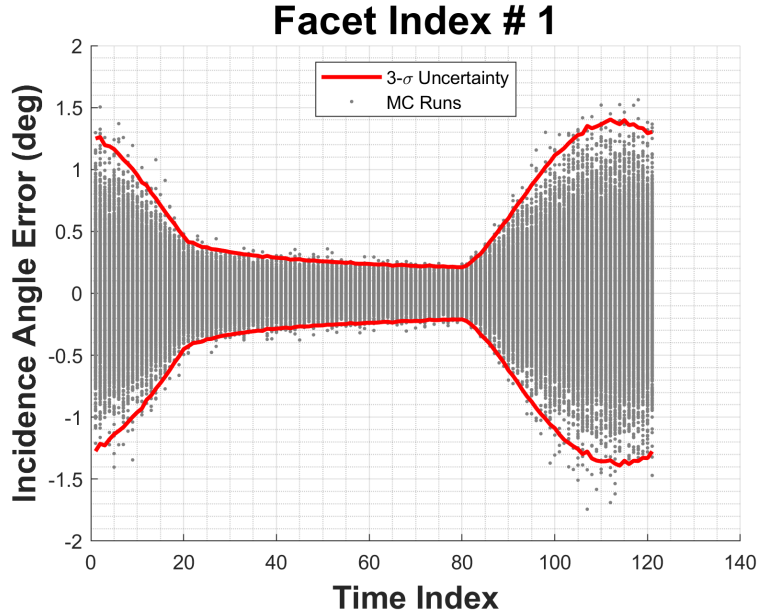


Figure 6.16: Incidence angle Monte Carlos (1000 runs) for facet 1 of the shape model. Analytical uncertainty ($3\text{-}\sigma$) bounds shown in red

the analytically derived uncertainty (Eq. (6.82)). The analytical covariance is derived by linear mapping of the measurement uncertainty into the state space. This linear mapping does not accurately capture the uncertainty as demonstrated by the Monte Carlo results. This suggests that the linear map is insufficient and more advanced techniques such as the unscented transform [55, 56] are necessary to accurately determine the uncertainties in the state.

6.6 Summary

In this chapter, a new algorithm called stereo-thermoclinometry (STC) is presented. STC uses temperature measurements from infrared images and predicted measurements from a Thermo-Physical Model (TPM) to refine the shape model of a small body given an initial guess. Unlike silhouette based methods (such as the one outlined in Chapter 5), STC is able to characterize concavities. STC can be summarized in four key steps that are iterated until convergence: (1) predicted temperatures using a TPM (2) pixel to facet mapping (Section 6.2.2) (2) thermoclinometry (Section 6.2.3) and (2) vertex optimization (Section 6.2.4). Measured temperatures from pixels

Facet Index # 1

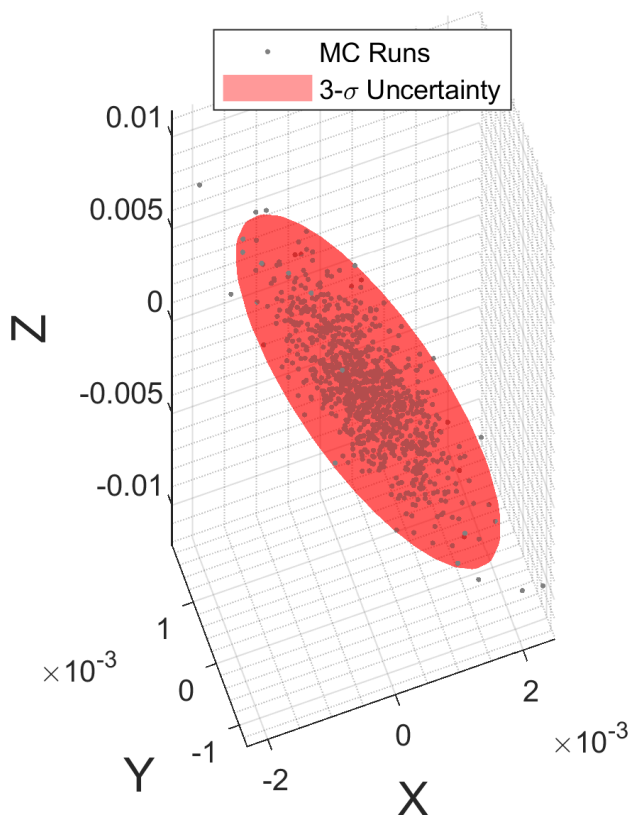


Figure 6.17: Facet normal Monte Carlo (1000 runs) for facet 1 of the shape model. Analytical uncertainty (3- σ) ellipse shown in red

Vertex Index # 1

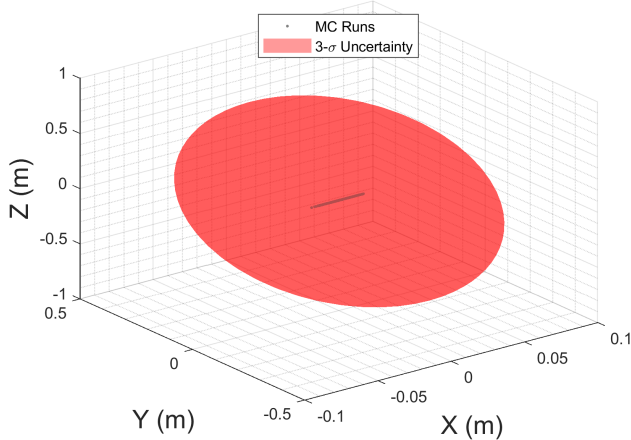


Figure 6.18: Monte Carlo results (1000 runs) for vertex 1 of the shape model. Analytical uncertainty (3- σ) ellipse shown in red

of an infrared image are associated with a facet on the initial shape model to generate measured temperatures of each facet. The thermoclinometry process estimates the facet orientations from the predicted and measured facet temperatures. These orientations are then used to update the vertices of the shape model using a Quasi-Newton optimization algorithm. These steps are iterated until convergence.

The algorithm is applied to images of a spherical small body with a crater with a depth of 2.3 m and a diameter of 33.6 m. In a baseline case where the observer is 20 km from the center of mass (CoM) of the small body along a latitude and longitude of 0° , the STC derived shape model (after 60 iterations) achieves a 93 % reduction in error in comparison to the initial shape model. At this distance, the crater spans approximately 20 pixels along both the horizontal and vertical axis of the image. Analysis of different observational latitudes and longitudes is performed. This shows that convergence is slower and the final shape error is higher for latitudes and longitudes that are further away from 0° . This performance degradation is caused by the projection errors inherent to the mapping process. However, the algorithm is able to reduce the error by a significant 66 % in the worst case (at -60° latitude and -50° longitude). Sensitivity of the algorithm to the axial tilt and to image frequency is performed. The algorithm's performance is better when the magnitude of the axial tilt $|\gamma|$ is larger. Furthermore, analysis shows that image/measurement frequency isn't as important as measurement coverage across the spin period. The algorithm requires very few images (as few as 11) to converge and the performance is very similar across different number of images (between 11 and 121 images). However, sufficient coverage of the day/night cycle is necessary to produce accurate characterization of the surface orientations. This coverage is achieved by ensuring that both day and night time measurements are used in the thermoclinometry step of STC. STC is also applied to images taken in a flyby trajectory in which a spacecraft approaches the body along the terminator plane (plane orthogonal to the sun direction). The converged shape model has a high accuracy which demonstrates the effectiveness of this algorithm in a scenario where observational geometry is limited. Further analysis of the algorithm behavior needs to be conducted to understand performance across different initial small body shape models, morphologies, and

observational geometries. A discussion on avenues for further development of the STC algorithm is given Chapter [7](#).

Chapter 7

Conclusion

In this thesis, we present algorithms that utilize infrared images for shape modeling and pole estimation of small bodies. Chapter 1 outlines the background on small body exploration strategies including the sensors and current algorithms used to extract pole and shape information. Chapter 2 outlines the relevant frame definitions, thermal models, and image rendering techniques utilized in the thesis.

Chapter 3 outlines a method to extract the pole axis of a small body from a set of on-board infrared images. The geometry between the spin pole axis and the camera frame is observable in a set of infrared images. The algorithm exploits this observable geometry to obtain an estimate of the pole axis. Results on synthetic images of asteroids Bennu and Itokawa show good performance at equatorial observation latitudes while the polar observation latitudes shows relatively worse performance. The sensitivity of the ellipse fitting algorithm used to extract the geometry has a significant impact on the resultant pole error. Nevertheless, these errors are less than the ground-based pole estimates derived using lightcurve inversion of photometric in most cases. The algorithm is also tested on real images of asteroid Ryugu which results in a 9° error. Analysis on sensitivity to pointing and navigation errors should be performed to more comprehensively characterize the performance of the algorithm. One could also utilize the initial estimate of the pole axis to bootstrap an iterative ellipse fitting algorithm to further reduce the errors.

Similar geometry is observed between the orbit pole of a binary asteroid system and the camera frame. In Chapter 4, an algorithm is outlined to estimate the orbit pole from camera

observations of the mutual orbit. If the mutual orbit is assumed to be circular (*i.e.* eccentricity is 0), the algorithm is able to estimate the orbit pole to within 10° in most cases (even when the true eccentricity is not 0). A potential direction of research of this algorithm is to use the initial estimate derived under the circular assumption to get a better estimate of the orbit pole axis and eccentricity.

Chapter 5 outlines an algorithm to characterize small body shapes using the silhouettes from a set of infrared observations. The silhouette is a robust feature that can be reliably extracted from a set of images once the body is resolved (spans more than a pixel). Infrared images allow for light-robust extraction of this quantity as the entire silhouette is always visible regardless of the solar phase angle. A novel sampling method is presented which extracts the salient silhouette pixels such that accurate shape model performance is achieved with a lower computational cost. The algorithm performance is characterized on synthetic images of Itokawa, Bennu, Eros and 67PC-G which show good performance with RMS of 0.4609%, 1.0303%, 2.038%, 1.938% of the respective body radii. Analysis shows that combining images from different latitudes results in greater performance as this can overcome observability limitations of any given image. The algorithm is tested on real images of asteroid Ryugu and the resultant shape model has an RMS error of 1.640% of the body radius. In the algorithm, the center of mass (CoM) is assumed to be known. While the center of brightness (CoB) of infrared images correlates well with the true CoM (unlike high solar phase angle optical images), estimating this quantity would improve the robustness of the shape modeling algorithm.

The SfS derived shape models are unable to characterize certain shape morphologies (*e.g.* craters) as these do not appear on the silhouettes. Thus, in Chapter 6, an algorithm called stereo-thermoclinometry (STC) is proposed to refine the initial silhouette based model. The algorithm estimates the surface orientations from a set of temperature measurements in a process called *thermoclinometry* and then optimizes the shape vertices based on these estimates. Sensitivities of the algorithm to observational geometry, axial tilt, and image frequency is performed. The observational geometry study shows better convergence when the camera plane is parallel to the feature (*e.g.* crater) plane. This study reveals the algorithm's sensitivity to the axial tilt and

latitude where the feature exists. When these two quantities are closer each other, the errors are larger and when they are further apart, the errors are lower. The image frequency analysis showed that there is negligible difference in performance by using as few as 11 or as many as 121 images as long as these measurements are spread across the day/night cycle of the asteroid. Concurrent estimation of the thermal parameters along with the surface orientations would alleviate the need for accurate prior estimates of the thermal parameters. In doing so, thermal inertia variation across the surface can also be estimated. Combining structure from motion techniques with STC could allow for alleviate the projection issues as it results in a better initial shape model.

These algorithms allow for computationally efficient and illumination robust characterization of the spin pole axis and shape of small bodies through the novel use of infrared images. As infrared cameras have been useful scientific instruments, no additional hardware impositions are made in order to facilitate their use for autonomous small body characterization. The autonomous design of these algorithms enhances mission autonomy by reducing the reliance on ground teams for processing the data. These technologies enable missions of a lower cost class such as NASA's SIMPLEx¹ (Small Innovative Missions for PLanetary Exploration) class [113] which leads to greater number of explored bodies for the same cost as an expensive class mission. By doing so, comprehensive characterization of more bodies is achieved which leads to improved knowledge of planetary formation, more informed planetary defense capabilities, and may be the key to sustained exploration of the solar system through enabling in-situ resource utilization.

¹ [NASA SIMPLEx Website](#)

Bibliography

- [1] S. Abe, T. Mukai, N. Hirata, O. S. Barnouin-Jha, A. F. Cheng, H. Demura, R. W. Gaskell, T. Hashimoto, K. Hiraoka, T. Honda, T. Kubota, M. Matsuoka, T. Mizuno, R. Nakamura, D. J. Scheeres, and M. Yoshikawa. Mass and local topography measurements of itokawa by hayabusa. Science, 312(5778):1344–1347, 2006.
- [2] C. Acton, N. Bachman, B. Semenov, and E. Wright. A look towards the future in the handling of space science mission geometry. Planetary and Space Science, 150:9–12, 1 2018.
- [3] C. H. Acton. Ancillary data services of nasa’s navigation and ancillary information facility. Planetary and Space Science, 44:65–70, 1 1996.
- [4] S. J. Ahn, W. Rauh, and H. J. Warnecke. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. Pattern Recognition, 34:2283–2303, 12 2001.
- [5] A. Akula, R. Ghosh, S. Kumar, and H. Sardana. Wignermser: Pseudo-wigner distribution enriched mser feature detector for object recognition in thermal infrared images. IEEE Sensors Journal, 19(11):4221–4228, 2019.
- [6] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. Kaze features. In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12, pages 214–227. Springer, 2012.
- [7] H. A. AlMazmi, P. O. Hayne, M. E. Landis, and W. F. Bottke. Science Overview of the Emirates Mission to the Asteroid Belt. In AGU Fall Meeting Abstracts, volume 2023, pages P44B–01, Dec. 2023.
- [8] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, SMA '01, page 249–266, New York, NY, USA, 2001. Association for Computing Machinery.
- [9] I. Amidror. Scattered data interpolation methods for electronic imaging systems: a survey. Journal of electronic imaging, 11(2):157–176, 2002.
- [10] T. Arai, T. Nakamura, S. Tanaka, H. Demura, Y. Ogawa, N. Sakatani, Y. Horikawa, H. Sen-shu, T. Fukuhara, and T. Okada. Thermal imaging performance of tir onboard the hayabusa2 spacecraft. Space Science Reviews, 208:239–254, 7 2017.
- [11] L. Armijo. Minimization of functions having lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1):1–3, 1966.

- [12] D. Arthur, S. Vassilvitskii, et al. k-means++: The advantages of careful seeding. In Soda, volume 7, pages 1027–1035, 2007.
- [13] V. Badescu. Asteroids: Prospective energy and material resources. Springer Science & Business Media, 2013.
- [14] D. Baker and J. McMahon. Limb-based shape modeling and localization for autonomous navigation around small bodies. In 2020 Astrodynamics Specialist Conference, Lake Tahoe, 2020.
- [15] S. Bandyonadhyay, I. Nesnas, S. Bhaskaran, B. Hockman, and B. Morrell. Silhouette-based 3d shape reconstruction of a small body from a spacecraft. In 2019 IEEE Aerospace Conference, pages 1–13. IEEE, 2019.
- [16] O. Barnouin, M. Daly, E. Palmer, R. Gaskell, J. Weirich, C. Johnson, M. Al Asad, J. Roberts, M. Perry, H. Susorney, et al. Shape of (101955) bennu indicative of a rubble pile with internal stiffness. Nature geoscience, 12(4):247–252, 2019.
- [17] O. Barnouin, M. Daly, E. Palmer, R. Gaskell, J. Weirich, C. Johnson, M. Al Asad, J. Roberts, M. Perry, H. Susorney, et al. Shape of (101955) bennu indicative of a rubble pile with internal stiffness. Nature geoscience, 12(4):247–252, 2019.
- [18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). Computer vision and image understanding, 110(3):346–359, 2008.
- [19] B. Bos, D. Nelson, J. Pelgrift, A. Liounis, D. Doelling, C. Norman, R. Olds, C. May, R. Witherspoon, E. Church, et al. In-flight calibration and performance of the osiris-rex touch and go camera system (tagcams). Space Science Reviews, 216:1–52, 2020.
- [20] J. Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6):679–698, 1986.
- [21] C. Capanna, G. Gesquière, L. Jorda, P. Lamy, and D. Vibert. Three-dimensional reconstruction using multiresolution photoclinometry by deformation. The Visual Computer, 29:825–835, 2013.
- [22] C. Capanna, L. Jorda, G. Gesquière, R. W. Gaskell, O. Groussin, P. Gutiérrez, S. F. Hviid, P. L. Lamy, F. Preusker, S. Rodionov, et al. Multi-resolution stereophotoclinometry by deformation, a new 3d shape reconstruction method applied to rosetta/osiris images. In AGU Fall Meeting Abstracts, volume 2014, pages P41C–3941, 2014.
- [23] K.-M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part i: Theory and algorithms. International Journal of Computer Vision, 62:221–247, 2005.
- [24] A. Y. S. Chia, M. K. Leung, H. L. Eng, and S. Rahardja. Ellipse detection with hough transform in one dimensional parametric space. Proceedings - International Conference on Image Processing, ICIP, 5, 2007.
- [25] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman. Novel quaternion kalman filter. IEEE Transactions on Aerospace and Electronic Systems, 42(1):174–190, 2006.

- [26] E. Christensen, S. Larson, A. Boattini, A. Gibbs, A. Grauer, R. Hill, J. Johnson, R. Kowalski, and R. McNaught. The catalina sky survey: current and future work. In AAS/Division for Planetary Sciences Meeting, volume 44, pages 210–13, 2012.
- [27] J. A. Christian, L. Benhacine, J. Hikes, and C. D’Souza. Geometric calibration of the orion optical navigation camera using star field images. The Journal of the Astronautical Sciences, 63:335–353, 2016.
- [28] C. F. Chyba, P. J. Thomas, and K. J. Zahnle. The 1908 tunguska explosion: atmospheric disruption of a stony asteroid. Nature, 361(6407):40–44, 1993.
- [29] G. L. Civardi, M. Bechini, M. Quirino, A. Colombo, M. Piccinin, and M. Lavagna. Generation of fused visible and thermal-infrared images for uncooperative spacecraft proximity navigation. Advances in Space Research, 73(11):5501–5520, 2024.
- [30] G. L. Civardi, M. Piccinin, M. Lavagna, et al. Small bodies ir imaging for vision based relative navigation and mapping enhancement. In 7th IAA Planetary Defense Conference (PDC 2021), pages 1–15, 2021.
- [31] A. Crida. Solar system formation. Reviews in Modern Astronomy: Formation and Evolution of Cosmic Structures, pages 215–227, 2009.
- [32] C. Cunningham. Discovery of the first asteroid, ceres. Historical studies in asteroid research, 2016.
- [33] M. Delbo, M. Mueller, J. Emery, B. Rozitis, and M. Capria. Asteroid thermophysical modeling. Asteroids IV, 08 2015.
- [34] P. Denchev, P. Magnusson, and Z. Donchev. Lightcurves of nine asteroids, with pole and sense of rotation of 42 isis. Planetary and Space Science, 46:673–682, 1998.
- [35] C. M. Ernst, R. Gaskell, R. T. Daly, and O. S. Barnouin. A stereophotoclinometry model of comet tempel 1. In AAS/Division for Planetary Sciences Meeting Abstracts# 50, volume 50, pages 112–01, 2018.
- [36] A. Fitzgibbon, M. Pilu, and R. Fisher. Direct least square fitting of ellipses. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5):476–480, 1999.
- [37] R. Fletcher. Practical methods of optimization. John Wiley & Sons, 2000.
- [38] M. Fornaciari, A. Prati, and R. Cucchiara. A fast and effective ellipse detector for embedded vision applications. Pattern Recognition, 47:3693–3708, 2014.
- [39] E. Galimov, V. Kolotov, M. Nazarov, Y. A. Kostitsyn, I. Kubrakova, N. Kononkova, I. Roshchina, V. Alexeev, L. Kashkarov, D. Badyukov, et al. Analytical results for the material of the chelyabinsk meteorite. Geochemistry International, 51:522–539, 2013.
- [40] R. Gaskell, O. Barnouin-Jha, and D. Scheeres. Modeling eros with stereophotoclinometry. In 38th Annual Lunar and Planetary Science Conference, number 1338, page 1333, 2007.
- [41] R. Gaskell, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, N. Hirata, S. Abe, O. Barnouin-Jha, and D. Scheeres. Gaskell itokawa shape model v1. 0. NASA Planetary Data System, pages HAY–A, 2008.

- [42] R. W. Gaskell, O. S. Barnouin, M. G. Daly, E. E. Palmer, J. R. Weirich, C. M. Ernst, R. T. Daly, and D. S. Lauretta. Stereophotoclinometry on the osiris-rex mission: Mathematics and methods. *The Planetary Science Journal*, 4:63, 4 2023.
- [43] R. W. Gaskell, O. S. Barnouin-Jha, D. Scheeres, A. Konopliv, T. Mukai, S. Abe, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, J. Kawaguchi, M. Yoshikawa, K. Shirakawa, T. Kominato, N. Hirata, and H. Demura. Characterizing and navigating small bodies with imaging data. *Meteoritics & Planetary Science*, 43:1049–1061, 6 2008.
- [44] S. Ghosal and R. Mehrotra. Orthogonal moment operators for subpixel edge detection. *Pattern recognition*, 26(2):295–306, 1993.
- [45] M. W. Givens, **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Autonomous cislunar navigation using a visual direction of motion measurement and a gaussian mixture model filter. In *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, USA, August 2023.
- [46] K.-H. Glassmeier, H. Boehnhardt, D. Koschny, E. Kürt, and I. Richter. The rosetta mission: flying towards the origin of the solar system. *Space Science Reviews*, 128:1–21, 2007.
- [47] D. Golish, C. Drouet d’Aubigny, B. Rizk, D. DellaGiustina, P. Smith, K. Becker, N. Shultz, T. Stone, M. Barker, E. Mazarico, et al. Ground and in-flight calibration of the osiris-rex camera suite. *Space science reviews*, 216:1–31, 2020.
- [48] J. Hanuš, J. Ďurech, M. Brož, B. D. Warner, F. Pilcher, R. Stephens, J. Oey, L. Bernasconi, S. Casulli, R. Behrend, D. Polishook, T. Henych, M. Lehký, F. Yoshida, and T. Ito. A study of asteroid pole-latitude distribution based on an extended set of shape models derived by the lightcurve inversion method. *Astronomy & Astrophysics*, 530:A134, 6 2011.
- [49] E. F. Helin, S. H. Pravdo, D. L. Rabinowitz, and K. J. Lawrence. Near-earth asteroid tracking (neat) program. *Annals of the New York Academy of Sciences*, 822(1):6–25, 1997.
- [50] C. W. Hergenrother, C. K. Maleszewski, M. C. Nolan, and The OSIRIS-REx Team. The operational environment and rotational acceleration of asteroid (101955) bennu from osiris-rex observations. *Nature Communications* 2019 10:1, 10:1–10, 3 2019.
- [51] T. Hermosilla, E. Bermejo, A. Balaguer, and L. A. Ruiz. Non-linear fourth-order image interpolation for subpixel edge detection and localization. *Image and vision computing*, 26(9):1240–1248, 2008.
- [52] D. Jewitt. The active asteroids. *The Astronomical Journal*, 143(3):66, 2012.
- [53] D. Jewitt, H. Hsieh, J. Agarwal, et al. The active asteroids. *Asteroids IV*, pages 221–241, 2015.
- [54] J. Johansson, M. Solli, and A. Maki. An evaluation of local feature detectors and descriptors for infrared images. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part III 14*, pages 711–723. Springer, 2016.
- [55] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000.

- [56] S. J. Julier. The scaled unscented transformation. In Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), volume 6, pages 4555–4559. IEEE, 2002.
- [57] M. Kaasalainen, M. Abe, J. Byron, M. Delbo, M. Di Martino, D. Higgins, K. Kitazato, S. Lowry, G. Masi, M. Mueller, J. Näränen, D. Vokrouhlicky, B. Warner, P. Weissman, and J. Young. Photometric observations 2001-2004 and modelling of (25143) itokawa. 12 2004.
- [58] M. Kaasalainen, T. Kwiatkowski, M. Abe, J. Piironen, T. Nakamura, Y. Ohba, B. Dermawan, T. Farnham, F. Colas, S. Lowry, et al. Ccd photometry and model of muses-c target (25143) 1998 sf36. Astronomy & Astrophysics, 405(3):L29–L32, 2003.
- [59] M. Kaasalainen and J. Torppa. Optimization methods for asteroid lightcurve inversion: I. shape determination. Icarus, 153(1):24–36, 2001.
- [60] M. Kaasalainen, J. Torppa, and K. Muinonen. Optimization methods for asteroid lightcurve inversion: II. the complete inverse problem. Icarus, 153(1):37–51, 2001.
- [61] N. Kaiser, H. Aussel, B. E. Burke, H. Boesgaard, K. Chambers, M. R. Chun, J. N. Heasley, K.-W. Hodapp, B. Hunt, R. Jedicke, et al. Pan-starrs: a large synoptic survey telescope array. In Survey and Other Telescope Technologies and Discoveries, volume 4836, pages 154–164. SPIE, 2002.
- [62] M. Kanamaru, T. Okada, H. Senshu, H. Demura, N. Hirata, Y. Horikawa, and G. Tommei. Thermophysical model development for hera mission to simulate non-gravitational acceleration on binary asteroid. 9 2023.
- [63] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In Proceedings of the fourth Eurographics symposium on Geometry processing, volume 7, 2006.
- [64] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG), 32(3):1–13, 2013.
- [65] K. Kuppa, J. W. McMahon, and A. B. Dietrich. Initial pole axis and spin direction estimation of asteroids using infrared imagery. Journal of Guidance, Control, and Dynamics, 47(6):1055–1071, 2024.
- [66] S. Larson, J. Brownlee, C. Hergenrother, and T. Spahr. The catalina sky survey for neos. In American Astronomical Society, DPS meeting# 30, id. 12. P14; Bulletin of the American Astronomical Society, Vol. 30, p. 1037, volume 30, page 1037, 1998.
- [67] D. Lauretta, S. Balram-Knutson, E. Beshore, W. Boynton, C. Drouet d’Aubigny, D. DellaGiustina, H. Enos, D. Golish, C. Hergenrother, E. Howell, et al. Osiris-rex: sample return from asteroid (101955) bennu. Space Science Reviews, 212:925–984, 2017.
- [68] D. Lauretta, D. DellaGiustina, C. Bennett, D. Golish, K. Becker, S. Balram-Knutson, O. Barnouin, T. Becker, W. Bottke, W. Boynton, et al. The unexpected surface of asteroid (101955) bennu. Nature, 568(7750):55–60, 2019.
- [69] D. S. Lauretta, H. C. Connolly Jr, J. E. Aebersold, C. M. O. Alexander, R.-L. Ballouz, J. J. Barnes, H. C. Bates, C. A. Bennett, L. Blanche, E. H. Blumenfeld, et al. Asteroid (101955) bennu in the laboratory: Properties of the sample collected by osiris-rex. Meteoritics & Planetary Science, 2024.

- [70] H. F. Levison, C. B. Olkin, K. S. Noll, S. Marchi, J. F. Bell III, E. Bierhaus, R. Binzel, W. Bottke, D. Britt, M. Brown, et al. Lucy mission to the trojan asteroids: science goals. The Planetary Science Journal, 2(5):171, 2021.
- [71] Z. Y. Liu and H. Qiao. Multiple ellipses detection in noisy environments: A hierarchical approach. Pattern Recognition, 42:2421–2433, 11 2009.
- [72] P. Lord, S. Tilley, D. Y. Oh, D. Goebel, C. Polanskey, S. Snyder, G. Carr, S. M. Collins, G. Lantoine, D. Landau, et al. Psyche: Journey to a metal world. In 2017 IEEE Aerospace Conference, pages 1–11. IEEE, 2017.
- [73] D. G. Lowe. Object recognition from local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision, volume 2, pages 1150–1157. Ieee, 1999.
- [74] S. Lowry, S. Duddy, B. Rozitis, S. F. Green, A. Fitzsimmons, C. Snodgrass, H. H. Hsieh, and O. Hainaut. The nucleus of comet 67p/churyumov-gerasimenko-a new shape model and thermophysical analysis. Astronomy & Astrophysics, 548:A12, 2012.
- [75] E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves. Subpixel measurements using a moment-based edge operator. IEEE Transactions on pattern analysis and machine intelligence, 11(12):1293–1309, 1989.
- [76] E. M. MacLennan and J. P. Emery. Thermophysical modeling of asteroid surfaces using ellipsoid shape models. The Astronomical Journal, 157:2, 12 2018.
- [77] E. S. Maini. Enhanced direct least square fitting of ellipses. International Journal of Pattern Recognition and Artificial Intelligence, 20(06):939–953, 2006.
- [78] J.-L. Margot, M. Nolan, L. Benner, S. Ostro, R. Jurgens, J. Giorgini, M. Slade, and D. Campbell. Binary asteroids in the near-earth object population. Science, 296(5572):1445–1448, 2002.
- [79] I. Martin, M. Dunstan, D. Vural, and M. Sanchez Gestido. Thermal image simulation to support gnc testing with pangu (paper siw22-05). oct 2022. 3rd Space Imaging Workshop ; Conference date: 10-10-2022 Through 12-10-2022.
- [80] K. Matsumoto, H. Noda, Y. Ishihara, H. Senshu, K. Yamamoto, N. Hirata, N. Hirata, N. Namiki, T. Otsubo, A. Higuchi, et al. Improving hayabusa2 trajectory by combining lidar data and a shape model. Icarus, 338:113574, 2020.
- [81] A. McAndrew. An introduction to digital image processing with MATLAB. Course Technology Press, 2004.
- [82] W. J. Merline, S. J. Weidenschilling, D. D. Durda, J.-L. Margot, P. Pravec, and A. D. Storrs. Asteroids do have satellites. Asteroids III, 1:289–312, 2002.
- [83] P. Michel. Formation and physical properties of asteroids. Elements, 10(1):19–24, 2014.
- [84] F. Mignard, A. Cellino, K. Muinonen, P. Tanga, M. Delbo, A. Dell’Oro, M. Granvik, D. Hestroffer, S. Mouret, W. Thuillot, et al. The gaia mission: Expected applications to asteroid science. Earth, Moon, and Planets, 101:97–125, 2007.

- [85] J. K. Miller, A. Konopliv, P. Antreasian, J. Bordi, S. Chesley, C. Helfrich, W. Owen, T. Wang, B. Williams, D. Yeomans, et al. Determination of shape, gravity, and rotational state of asteroid 433 eros. Icarus, 155(1):3–17, 2002.
- [86] S. Mottola, S. Lowry, C. Snodgrass, P. Lamy, I. Toth, A. Rožek, H. Sierks, M. F. A’Hearn, F. Angrilli, C. Barbieri, et al. The rotation state of 67p/churyumov-gerasimenko from approach observations with the osiris cameras on rosetta. Astronomy & Astrophysics, 569:L2, 2014.
- [87] A. Y. Mulayim, U. Yilmaz, and V. Atalay. Silhouette-based 3-d model reconstruction from multiple images. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 33(4):582–591, 2003.
- [88] T. G. Müller, J. Ďurech, M. Ishiguro, M. Mueller, T. Krühler, H. Yang, M. J. Kim, L. O’Rourke, F. Usui, C. Kiss, B. Altieri, B. Carry, Y. J. Choi, M. Delbo, J. P. Emery, J. Greiner, S. Hasegawa, J. L. Hora, F. Knust, D. Kuroda, D. Osip, A. Rau, A. Rivkin, P. Schady, J. Thomas-Osip, D. Trilling, S. Urakawa, E. Vilenius, P. Weissman, and P. Zeidler. Hayabusa-2 mission target asteroid 162173 ryugu (1999 ju3): Searching for the object’s spin-axis orientation. Astronomy & Astrophysics, 599:A103, 3 2017.
- [89] F. Niklaus, C. Vieider, and H. Jakobsen. Mems-based uncooled infrared bolometer arrays: a review. MEMS/MOEMS technologies and applications III, 6836:125–139, 2008.
- [90] J. Nocedal and S. J. Wright. Numerical optimization. Springer, 1999.
- [91] M. C. Nolan, C. Magri, E. S. Howell, L. A. Benner, J. D. Giorgini, C. W. Hergenrother, R. S. Hudson, D. S. Lauretta, J. L. Margot, S. J. Ostro, and D. J. Scheeres. Shape model and surface properties of the osiris-rex target asteroid (101955) bennu from radar and lightcurve observations. Icarus, 226:629–640, 9 2013.
- [92] T. Okada, T. Fukuhara, S. Tanaka, M. Taguchi, T. Arai, H. Senshu, N. Sakatani, Y. Shimaki, H. Demura, Y. Ogawa, et al. Thermophysical properties of c-type asteroid 162173 ryugu revealed by the thermal infrared imager tir on hayabusa2. Transactions of The Japan Society For Aeronautical and Space Sciences, Aerospace Technology Japan, 19(5):654–659, 2021.
- [93] T. Okada, T. Fukuhara, S. Tanaka, M. Taguchi, T. Imamura, T. Arai, H. Senshu, Y. Ogawa, H. Demura, K. Kitazato, R. Nakamura, T. Kouyama, T. Sekiguchi, S. Hasegawa, T. Matsunaga, T. Wada, J. Takita, N. Sakatani, Y. Horikawa, K. Endo, J. Helbert, T. G. Müller, and A. Hagermann. Thermal infrared imaging experiments of c-type asteroid 162173 ryugu on hayabusa2. Space Science Reviews, 208:255–286, 7 2017.
- [94] T. Okada, S. Tanaka, T. Fukuhara, T. Arai, T. Imamura, Y. Ogawa, K. Kitazato, T. Kouyama, N. Sakatani, Y. Shimaki, T. Sekiguchi, H. Senshu, J. Takita, M. Taguchi, H. Demura, R. Nakamura, S. Hasegawa, T. Matsunaga, T. Wada, J. Helbert, T. G. Müller, A. Hagermann, J. Biele, M. Grott, M. Hamm, and M. Delbo. Hayabusa2 tir bundle v1.0. NASA Planetary Data System, 2021.
- [95] N. Ozaki, T. Yamamoto, F. Gonzalez-Franquesa, R. Gutierrez-Ramon, N. Pushparaj, T. Chikazawa, D. A. Dei Tos, O. Çelik, N. Marmo, Y. Kawakatsu, et al. Mission design of destiny+: Toward active asteroid (3200) phaethon and multiple small bodies. Acta Astronautica, 196:42–56, 2022.

- [96] E. E. Palmer, R. Gaskell, M. G. Daly, O. S. Barnouin, C. D. Adam, and D. S. Lauretta. Practical stereophotoclinometry for modeling shape and topography on planetary missions. *The Planetary Science Journal*, 3:102, 5 2022.
- [97] P. Panicucci, J. Lebreton, R. Brochard, E. Zenou, and M. Delpech. Vision-based estimation of small body rotational state. *Acta Astronautica*, 213:177–196, 2023.
- [98] R. S. Park, A. T. Vaughan, A. S. Konopliv, A. I. Ermakov, N. Mastrodemos, J. C. Castillo-Rogez, S. P. Joy, A. Nathues, C. A. Polanskey, M. D. Rayman, J. E. Riedel, C. A. Raymond, C. T. Russell, and M. T. Zuber. High-resolution shape model of ceres from stereophotoclinometry using dawn imaging data. *Icarus*, 319:812–827, 2 2019.
- [99] J. S. Parker, F. S. Al Hameli, J. Knittel, E. Pilinski, S. Baskar, P. Imler, and M. H. Almashjari. Proximity operations strategy for the emirates mission to explore the asteroid belt (ema). In *AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, CO, USA, August 2024.
- [100] S. Pfalzner, M. B. Davies, M. Gounelle, A. Johansen, C. Münker, P. Lacerda, S. P. Zwart, L. Testi, M. Triaelloff, and D. Veras. The formation of the solar system. *Physica Scripta*, 90(6):068001, 2015.
- [101] D. K. Prasad, M. K. Leung, and C. Quek. Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46:1449–1465, 5 2013.
- [102] P. Pravec, P. Fatka, D. Vokrouhlický, P. Scheirich, J. Ďurech, D. Scheeres, P. Kušnirák, K. Hornoch, A. Galád, D. Pray, et al. Asteroid pairs: a complex picture. *Icarus*, 333:429–463, 2019.
- [103] P. Pravec, P. Scheirich, P. Kušnirák, L. Šarounová, S. Mottola, G. Hahn, P. Brown, G. Esquerdo, N. Kaiser, Z. Krzeminski, et al. Photometric survey of binary near-earth asteroids. *Icarus*, 181(1):63–93, 2006.
- [104] L. Prockter, S. Murchie, A. Cheng, S. Krimigis, R. Farquhar, A. Santo, and J. Trombka. The near shoemaker mission to asteroid 433 eros. *Acta Astronautica*, 51(1-9):491–500, 2002.
- [105] M. D. Rayman, P. A. Chadbourne, J. S. Culwell, and S. N. Williams. Mission design for deep space 1: a low-thrust technology validation mission. *Acta astronautica*, 45(4-9):381–388, 1999.
- [106] D. T. Renshaw and J. A. Christian. Subpixel localization of isolated edges and streaks in digital images. *Journal of Imaging*, 6(5):33, 2020.
- [107] P. Ricaurte, C. Chilán, C. A. Aguilera-Carrasco, B. X. Vintimilla, and A. D. Sappa. Feature point descriptors: Infrared and visible spectra. *Sensors*, 14(2):3690–3701, 2014.
- [108] K. Righter, P. Abell, D. Agresti, E. L. Berger, A. S. Burton, J. S. Delaney, M. Fries, E. Gibson, M. K. Haba, R. Harrington, et al. Mineralogy, petrology, chronology, and exposure history of the chelyabinsk meteorite and parent body. *Meteoritics & Planetary Science*, 50(10):1790–1819, 2015.
- [109] S. D. Ross. Near-earth asteroid mining. *Space*, 2001:1–24, 2001.

- [110] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In 2011 International conference on computer vision, pages 2564–2571. Ieee, 2011.
- [111] C. M. Rumpf, H. G. Lewis, and P. M. Atkinson. Asteroid impact effects and their immediate hazards for human populations. Geophysical Research Letters, 44(8):3433–3440, 2017.
- [112] B. V. Sarli, M. Horikawa, C. H. Yam, Y. Kawakatsu, and T. Yamamoto. Destiny+ trajectory design to (3200) phaethon. The Journal of the Astronautical Sciences, 65:82–110, 2018.
- [113] D. J. Scheeres, J. W. McMahon, E. B. Bierhaus, J. Wood, L. A. M. Benner, C. M. Hartzell, P. Hayne, R. Jedicke, L. Le Corre, A. Meyer, S. Naidu, P. Pravec, M. Ravine, and K. Sorli. Janus: A NASA SIMPLEX mission to explore two NEO Binary Asteroids. In 7th IAA Planetary Defense Conference, page 55, Apr. 2021.
- [114] P. Schulte, L. Alegret, I. Arenillas, J. A. Arz, P. J. Barton, P. R. Bown, T. J. Bralower, G. L. Christeson, P. Claeys, C. S. Cockell, et al. The chicxulub asteroid impact and mass extinction at the cretaceous-paleogene boundary. Science, 327(5970):1214–1218, 2010.
- [115] J. Seabrook, M. Daly, O. Barnouin, C. Johnson, A. Nair, E. Bierhaus, W. Boynton, R. Espiritu, R. Gaskell, E. Palmer, et al. Global shape modeling using the osiris-rex scanning laser altimeter. Planetary and Space Science, 177:104688, 2019.
- [116] G. F. Serio, A. Manara, P. Sicoli, and W. F. Bottke. Giuseppe piazzi and the discovery of ceres. Asteroids III, 08 2002.
- [117] K. C. Sorli and P. O. Hayne. Thermophysical modeling of binary asteroids using a sophisticated 3d ray tracing approach. TherMoPS IV Abstract Book, page 34, 2023.
- [118] G. H. Stokes, J. B. Evans, H. E. Viggh, F. C. Shelly, and E. C. Pearce. Lincoln near-earth asteroid program (linear). Icarus, 148(1):21–28, 2000.
- [119] D. Sunday. Practical geometry algorithms. Independently published, 2021.
- [120] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13:1115–1138, 1991.
- [121] **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Initial pole estimation of small bodies on approach using infrared imagery. In 2022 AAS/AIAA Astrodynamics Specialist Conference, Charlotte, NC, USA, August 2022.
- [122] **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Initial pole estimation using infrared imagery. In 2022 Space Imaging Workshop, Atlanta, GA, USA, October 2022.
- [123] **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Binary asteroid orbit pole estimation from spacecraft imagery. In 2023 Asteroids, Comets, Meteors Conference, Flagstaff, AZ, USA, June 2023.
- [124] **K. Kuppa**, J. W. McMahon, and A. B. Dietrich. Light-robust and autonomous spin pole estimation and shape modeling of asteroids using infrared imagery. In 2024 AAS GN&C Conference, Breckenridge, CO, USA, February 2024.

- [125] **K. Kupp**a, J. W. McMahon, and A. B. Dietrich. Stereo thermoclinometry (stc): A method to characterize small body shapes using infrared images. In AAS/AIAA Astrodynamics Specialist Conference, Broomfield, CO, USA, August 2024.
- [126] **K. Kupp**a, J. W. McMahon, and A. B. Dietrich. Stereo thermoclinometry: A method to characterize small body shapes using infrared images. 2024 (*in prep*).
- [127] **K. Kupp**a, J. W. McMahon, and A. B. Dietrich. Autonomous shape modeling of small bodies using infrared image silhouettes. Acta Astronautica, 2024 (*submitted*).
- [128] **K. Kupp**a, J. W. McMahon, and A. B. Dietrich. Stereo thermoclinometry (stc): A method to characterize asteroid shapes using infrared images. In 2024 Draper Research Symposium, April 2023.
- [129] **K. Kupp**a, J. W. McMahon, and A. B. Dietrich. Towards autonomous pole estimation of small bodies. In 2023 Draper Research Symposium, April 2023.
- [130] P. Thomas, J. Joseph, B. Carcich, J. Veverka, B. Clark, J. Bell III, A. Byrd, R. Chomko, M. Robinson, S. Murchie, et al. Eros: Shape, topography, and slope processes. Icarus, 155(1):18–37, 2002.
- [131] V. C. Thomas, J. M. Makowski, G. M. Brown, J. F. McCarthy, D. Bruno, J. C. Cardoso, W. M. Chiville, T. F. Meyer, K. E. Nelson, B. E. Pavri, et al. The dawn spacecraft. The Dawn Mission to Minor Planets 4 Vesta and 1 Ceres, pages 175–249, 2012.
- [132] J. Torppa, M. Kaasalainen, T. Michałowski, T. Kwiatkowski, A. Kryszczyńska, P. Denchev, and R. Kowalski. Shapes and rotational properties of thirty asteroids from photometric data. Icarus, 164:346–383, 8 2003.
- [133] A. Trujillo-Pino, K. Krissian, M. Alemán-Flores, and D. Santana-Cedr s. Accurate subpixel edge location based on partial area effect. Image and vision computing, 31(1):72–90, 2013.
- [134] J. Villa, **K. Kupp**a, J. W. McMahon, and J. Knittel. Silhouette-based pole estimation and shape reconstruction of asteroid 269 justitia. In 2024 AAS GN&C Conference, Breckenridge, CO, USA, February 2024.
- [135] S. N. Ward and E. Asphaug. Asteroid impact tsunamis: a probabilistic hazard assessment. Icarus, 145(1):64–78, 2000.
- [136] S.-i. Watanabe, M. Hirabayashi, N. Hirata, N. Hirata, R. Noguchi, Y. Shimaki, H. Ikeda, E. Tatsumi, M. Yoshikawa, S. Kikuchi, et al. Hayabusa2 arrives at the carbonaceous asteroid 162173 ryugu—a spinning top-shaped rubble pile. Science, 364(6437):268–272, 2019.
- [137] S.-i. Watanabe, Y. Tsuda, M. Yoshikawa, S. Tanaka, T. Saiki, and S. Nakazawa. Hayabusa2 mission overview. Space Science Reviews, 208:3–16, 2017.
- [138] P. Wolfe. Convergence conditions for ascent methods. SIAM review, 11(2):226–235, 1969.
- [139] P. Wolfe. Convergence conditions for ascent methods. ii: Some corrections. SIAM review, 13(2):185–188, 1971.

- [140] T. Yada, M. Abe, T. Okada, A. Nakato, K. Yogata, A. Miyazaki, K. Hatakeda, K. Kumagai, M. Nishimura, Y. Hitomi, et al. Preliminary analysis of the hayabusa2 samples returned from c-type asteroid ryugu. Nature Astronomy, 6(2):214–220, 2022.
- [141] J. Ye, G. Fu, and U. P. Poudel. High-accuracy edge detection with blurred edge model. Image and Vision Computing, 23(5):453–467, 2005.
- [142] O. Yilmaz, N. Aouf, L. Majewski, M. Sanchez-Gestido, and G. Ortega. Using infrared based relative navigation for active debris removal. In 10th International ESA Conference on Guidance, Navigation and Control Systems, 2017.
- [143] R. Zanetti, M. Majji, R. H. Bishop, and D. Mortari. Norm-constrained kalman filtering. Journal of guidance, control, and dynamics, 32(5):1458–1465, 2009.
- [144] R. Zhang, L. Xu, Z. Yu, Y. Shi, C. Mu, and M. Xu. Deep-irtarget: An automatic target detector in infrared imagery using dual-domain feature extraction and allocation. IEEE Transactions on Multimedia, 24:1735–1749, 2021.

Appendix A

Determining Shadows using Ray-Tracing

To accurately render an image of a shape model of a small body (parameterized by its facets and vertices), it is necessary to determine which parts of the shape model are illuminated by the sun. For a convex shape, this can be trivially done by checking the dot product (Eq. (A.1)) between the surface element normal and the direction to the sun. Surface elements with a positive dot product ($s > 0$) are *illuminated* and surface elements with a zero or negative dot product $s \leq 0$ are *shadowed*:

$$s = \hat{\mathbf{n}} \cdot \hat{\mathbf{s}} \tag{A.1}$$

For complex shapes with concavities, this simple method does not hold as surface elements with a positive s may be *occluded* by other surface elements and thus be in shadow. To determine this, a *ray-tracing* approach is employed. Consider a shape model parameterized a set of vertices and corresponding triangular facets. For each triangular facet, Eq. (A.1) can be first used to determine the shadow or illuminated state of the facets in the absence of occlusions. For each illuminated facet, a line from of the constituent vertices $\mathbf{v}$ along the sun direction $\hat{\mathbf{s}}$ is drawn. A line-plane intersection algorithm (from [119]) is used to determine if the intersection between the line and plane spanned by *another* facet is within the facet. If *two* of the three vertices of a given facet intersect another facet, then it is shadowed ($s = 0$). An algorithm pseudocode for this process is given in Algorithm 5.

Algorithm 5 Compute Shadows

Given: $F, V, \hat{\mathbf{s}}$ **Result:** s

▷ Shadow state for each facet

```

1: for facet  $F_i$  in facets  $F$  do
2:    $s_i \leftarrow \hat{\mathbf{n}} \cdot \hat{\mathbf{s}}$  ▷ Eq. (A.1)
3:   if  $s_i > 0$  then ▷ Facet is illuminated
4:     for facet  $F_k$  in facets  $F \& \neq F_i$  do ▷ Loop through other facets
5:       for  $\mathbf{v}_j$  in facet  $F_i$  do ▷ Each vertex in facet
6:          $\ell = \mathbf{v}_j + \eta \hat{\mathbf{s}}$  ▷ Parameterization of a Line
7:         if  $\text{intersect}(\ell, F_k)$  then ▷ If line intersects a facet using algorithm from Ref. [119]
8:            $\text{int} \leftarrow \text{int} + 1$ 
9:         end if
10:        if  $\text{int} > 2$  then ▷ All 3 vertices intersect facet  $F_j$ 
11:           $s_i = 0$  ▷ Set shadow state to 0
12:        end if
13:      end for
14:    end for
15:  end if
16: end for

```

Appendix B

Limb-Limb Intersection Procedure

This section outlines the polygon intersection procedure used to compute the intersection points between two limb sections. First, compute the normal vectors (Eq. (5.15)) of the two limb sections (quadrilaterals). If the normal vectors align (cross product is zero), then the limbs are in the same plane and therefore no intersection is defined to exist. Otherwise, compute the intersecting points of each line segment from Q_1 with the limb section Q_2 (and vice-versa) (using a line segment intersection procedure from Ref. [119]). From these intersecting points, the *valid* points are those that lie in both limb sections. These points are identified by checking if they are within the span of the limb section. This process will result in 1 or 2 points of intersection. A pseudocode for this procedure is given in Algorithm 6 which identifies the valid intersection points $\mathbf{i}_v$ from two limb sections Q_1, Q_2 .

Algorithm 6 Limb-Limb Intersection

Input: Q_1, Q_2

Output: $\mathbf{i}_v$ (Valid Intersection points)

```

1:  $\hat{\mathbf{n}}_1 \leftarrow \text{computeNormal}(Q_1)$  ▷ Eq. (5.15)
2:  $\hat{\mathbf{n}}_2 \leftarrow \text{computeNormal}(Q_2)$  ▷ Eq. (5.15)
3: if  $\|\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2\| > 0$  then ▷ Limbs not in the same plane
4:    $\mathbf{i}_1 \leftarrow \text{intersectWithLimbs}(Q_1, Q_2)$  ▷ Use [119]
5:    $\mathbf{i}_2 \leftarrow \text{intersectWithLimbs}(Q_2, Q_1)$  ▷ Use [119]
6:    $\mathbf{I} \leftarrow [\mathbf{i}_1, \mathbf{i}_2]$ 
7:   for intersection points  $\mathbf{i}$  in  $\mathbf{I}$  do ▷ Check which points lie in both limbs
8:     if  $\text{isInLimb}(\mathbf{i}, Q_1) \ \& \ \text{isInLimb}(\mathbf{i}, Q_2)$  then
9:        $\mathbf{i}_v \leftarrow \mathbf{i}$  ▷ Append valid intersection
10:    end if
11:  end for
12: end if
```

Appendix C

Facet Normal Sensitivity to Vertices

In this section, the sensitivity of a facet normal $\hat{\mathbf{n}}$ w.r.t. its vertices $\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l$, is derived.

Recall the definition of the facet normal $\hat{\mathbf{n}}$:

$$\hat{\mathbf{n}} = \frac{\mathbf{a}_i}{\|\mathbf{a}\|} \quad (\text{C.1})$$

$$\mathbf{a} = (\mathbf{v}_k - \mathbf{v}_j) \times (\mathbf{v}_l - \mathbf{v}_j) \quad (\text{C.2})$$

The derivative of the normal w.r.t the vertex $\mathbf{v}$ is given by:

$$\frac{\partial \hat{\mathbf{n}}}{\partial \mathbf{v}} = \left(\frac{\partial \mathbf{a}}{\partial \mathbf{v}} \|\mathbf{a}\| - \frac{\partial \|\mathbf{a}\|}{\partial \mathbf{v}} \mathbf{a}^T \right) \frac{1}{\|\mathbf{a}\|^2} \quad (\text{C.3})$$

Let $u = \mathbf{a}^T \mathbf{a}$ such that:

$$\frac{\partial \|\mathbf{a}\|}{\partial \mathbf{v}} = \frac{\partial \sqrt{u}}{\partial \mathbf{v}} \quad (\text{C.4})$$

Applying the chain rule of differentiation:

$$\frac{\partial \sqrt{u}}{\partial \mathbf{v}} = \frac{1}{2} u^{-\frac{1}{2}} \frac{\partial u}{\partial \mathbf{v}} \quad (\text{C.5})$$

$$\frac{\partial u}{\partial \mathbf{v}} = \frac{\partial \mathbf{a}^T \mathbf{a}}{\partial \mathbf{v}} = 2 \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \mathbf{a} \quad (\text{C.6})$$

$\implies$

$$\frac{\partial \|\mathbf{a}\|}{\partial \mathbf{v}} = \frac{1}{\|\mathbf{a}\|} \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \mathbf{a} \quad (\text{C.7})$$

Thus, Eq. (C.3) can be expressed as:

$$\frac{\partial \hat{\mathbf{n}}}{\partial \mathbf{v}} = \left(\frac{\partial \mathbf{a}}{\partial \mathbf{v}} \|\mathbf{a}\| - \frac{1}{\|\mathbf{a}\|} \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \mathbf{a} \mathbf{a}^T \right) \frac{1}{\|\mathbf{a}\|^2} \quad (\text{C.8})$$

$$= \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \left(I_{3 \times 3} \frac{1}{\|\mathbf{a}\|} - \frac{1}{\|\mathbf{a}\|^3} \mathbf{a} \mathbf{a}^T \right) \quad (\text{C.9})$$

$$\frac{\partial \hat{\mathbf{n}}}{\partial \mathbf{v}} = \frac{1}{\|\mathbf{a}\|} \frac{\partial \mathbf{a}}{\partial \mathbf{v}} \left(I_{3 \times 3} - \frac{\mathbf{a} \mathbf{a}^T}{\|\mathbf{a}\|^2} \right) \quad (\text{C.10})$$

$\frac{\partial \mathbf{a}}{\partial \mathbf{v}}$ for the three vertices of a facet are given by:

$$\frac{\partial \mathbf{a}}{\partial \mathbf{v}_j} = [\mathbf{v}_l - \mathbf{v}_k]_{\times} \quad (\text{C.11})$$

$$\frac{\partial \mathbf{a}}{\partial \mathbf{v}_k} = [\mathbf{v}_j - \mathbf{v}_l]_{\times} \quad (\text{C.12})$$

$$\frac{\partial \mathbf{a}}{\partial \mathbf{v}_l} = [\mathbf{v}_k - \mathbf{v}_j]_{\times} \quad (\text{C.13})$$