

Copyright
by
John Allen Christian, III
2010

The Dissertation Committee for John Allen Christian, III
certifies that this is the approved version of the following dissertation:

Optical Navigation for a Spacecraft in a Planetary System

Committee:

E. Glenn Lightsey, Supervisor

Wallace Folwer

Belinda Marchand

Bob Schutz

Christopher D'Souza

**Optical Navigation for a Spacecraft in a Planetary
System**

by

John Allen Christian, III, B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2010

To my wife, Laura.

Acknowledgments

I would like to begin by thanking my supervisor, Dr. E. Glenn Lightsey. Thank you for providing me the freedom and opportunity to pursue the research that interests me. Thank you for your guidance, insight, advice, and critique of my work. I could not ask for a better supervisor and mentor.

I would also like to thank the members of my committee: Dr. Wallace Fowler, Dr. Belinda Marchand, Dr. Bob Schutz, and Dr. Christopher D'Souza. Thank you for taking the time to review this manuscript, answer my questions, and provide valuable feedback.

Thank you to Lisa Guerra for many thoughtful discussions over the past few years and for comments on parts of this manuscript. I would also like to thank my fellow graduate students at UT-Austin for making this process a little more bearable. In particular, I am especially grateful to Sebastian Muñoz, Dax Garner, Henri Kjellberg, Cinnamon Wright, Michael Davies, Serena Zhang, and Sara Scarritt.

Thank you to Robert Braun, my advisor as a master's student in the Space Systems Design Lab at the Georgia Institute of Technology. I also owe a special thanks to my fellow SSDL graduate students: Michael Grant, Chris Tanner, Brad Steinfeldt, John Theisinger, and Jarret Lafleur. Thank you for all you have done over the years - even after I left Georgia Tech.

I will be forever grateful to NASA, the Johnson Space Center, and the men and women that work there for all that they did for me in the 8 semesters (1 spring, 1 fall, 6 summers) that I was a co-op. The experience I gained through the opportunities they provided was critical to my professional growth. This dissertation is as much a testament to what I learned at JSC as to what I learned in school. A few individuals deserve special recognition. First and foremost, I would like to thank the three co-op coordinators that I have worked under: Bob Musgrove, Anne Roemer, and Tammie Wright. Additionally, I would like to thank a number of my previous colleagues and mentors who have helped me along the way, including: Rich Mrozinski, Robert Gay, Bret Drake, John Connolly, Joyce Carpenter, Jonathan Lenius, Brian Derkowski, Tim Fisher, Joe Trevathan, and Eric Christiansen.

Thank you to the MESSENGER science team for making mission data accessible through the NASA Planetary Data System. Numerous parts of this dissertation, including the Venus fly-by case study, would not have been possible without this data. I am also extremely grateful to the following organizations for granting me permission to reuse a number of images that greatly contributed to the completeness of this dissertation (especially in the background section): the American Institute of Aeronautics and Astronautics, Springer Science+Business Media (Space Science Reviews), SPIE, and the American Astronautical Society. I would also like to thank a number of individuals who gave me permission to reuse images from their previous work. In particular, I extend a special thanks to Arno Ledebuhr, Andrew Lowman, and John

Stauder. Additionally, thank you to Emil Schiesser for answering my many questions and helping me track down hard-to-find Apollo documents.

Thank you to my family. To my parents, John and Mary Beth Christian, thank you for a lifetime of love, support, and encouragement in whatever I chose to do. Thank you to my brothers, Bryan, Mark, and Dan, for always being there for me. To my parents-in-law, Paul and Karen Manno, and to my brother-in-law, Kevin, for their consistent encouragement.

Last, and certainly not least, I thank my wife, Laura. You shared the ups and downs of the last five years of graduate school with me and always made our home a place where I could forget the worries of being a graduate student. Thank you for always being patient and understanding when I had to spend my evenings or weekends studying for the qualifying exams or writing this dissertation. I could not have done this without your support. All I can say is thank you, and I love you.

I consider myself truly blessed. To each person mentioned above, and to the many other individuals that space prevents me from listing, I will always be grateful. None of this would have been possible without you.

JOHN A CHRISTIAN

The University of Texas at Austin

April 2010

Optical Navigation for a Spacecraft in a Planetary System

Publication No. _____

John Allen Christian, III, Ph.D.
The University of Texas at Austin, 2010

Supervisor: E. Glenn Lightsey

Recent years have seen ambitious robotic exploration missions to other planets and a renewed interest in sending humans beyond low Earth orbit. These activities give rise to a need for autonomous spacecraft operation. Of particular interest here is the ability of a spacecraft to navigate independent of contact with Earth-based resources. Optical navigation techniques are proposed as a solution to the problem of navigating in a planetary system without requiring navigation information from Earth. A detailed discussion of optical sensor hardware and error sources leads to new high fidelity math models for optical sensor performance that may be used in navigation simulations. Algorithms are developed that allow optical data to be used for the estimation of spacecraft position, velocity, and attitude. Sequential measurements are processed using traditional filtering techniques. Additionally, for the case of attitude estimation, a new attitude filter called Sequential Optimal Attitude Routine (SOAR) is presented. The models and techniques developed in

this dissertation are demonstrated in two case studies: (1) navigation of a spacecraft performing a planetary fly-by using real images from the June 2007 MESSENGER fly-by of Venus and (2) navigation of a spacecraft in cislunar space on a return trajectory from the Moon.

Table of Contents

Acknowledgments	v
Abstract	viii
List of Tables	xvii
List of Figures	xix
Chapter 1. Introduction	1
1.1 Historical Perspective and Motivation	2
1.1.1 Interplanetary Robotic Spaceflight	2
1.1.2 Human Spaceflight	4
1.2 Preliminary Navigation Considerations	7
1.3 Inertial Navigation	8
1.4 Sources of Autonomous External Navigation Updates	9
1.5 Contributions	11
1.5.1 Optical Sensor Model	12
1.5.2 Image Processing	12
1.5.3 The Sequential Optimal Attitude Recursion (SOAR) Filter	13
1.5.4 End-to-end Optical Navigation Performance Assessment	13
Chapter 2. Historical Background on Optical Navigation	15
2.1 Extension of Terrestrial Optical Navigation to Space Applications	15
2.2 Optical Navigation in Human Spaceflight	16
2.2.1 Gemini	17
2.2.2 Apollo	20
2.2.3 Skylab	23
2.2.4 Space Shuttle	23
2.3 Optical Navigation in Robotic Spacecraft	25

2.3.1	The Mariner Missions to Mars	25
2.3.2	Voyager and Cassini	27
2.3.3	Galileo	32
2.3.4	Near Earth Asteroid Rendezvous (NEAR) Mission . . .	39
2.3.5	Deep Space 1 (DS1)	40
2.3.6	MESSENGER	44
2.3.7	Robotic Missions to Mars	46
2.3.8	Robotic Missions to the Moon	47
Chapter 3. Image Creation and Optical Sensors		50
3.1	Rotations, Transformations, and Reference Frames	50
3.1.1	Basic Attitude Representations	50
3.1.2	Homogeneous Coordinates	53
3.1.3	Discussion of Coordinate Frames	54
3.2	Formal Definition of an Image	57
3.3	The Ideal Camera Model	58
3.4	Comments on Detector Hardware	65
3.4.1	Charged-Coupled Devices (CCD)	65
3.4.2	Complementary Metal Oxide Semiconductors (CMOS) .	68
3.4.3	CCD/CMOS Quantum Efficiency and Fill Factor	70
3.5	Overview of External Error Sources	71
3.5.1	Stellar Aberration	71
3.5.2	Parallax	74
3.5.3	Stray Light	76
3.6	Overview of Internal Camera Error Sources: Alignment & Optics	80
3.6.1	Camera Mounting Misalignment	80
3.6.2	Detector Plane Misalignment	81
3.6.3	Optical Aberrations	85
3.6.4	Diffraction-Limited Spot Size in a Focused Image	87
3.6.5	Point-Source Errors in a Defocused Image	88
3.7	Overview of Internal Camera Error Sources: CCD and CMOS Detectors	88
3.7.1	Dark Current and Shot Noise	89

3.7.2	Amplifier Noise	91
3.7.3	Reset Noise	92
3.7.4	Pattern Noise	93
3.7.5	ADC Quantization Noise	93
3.8	Camera Calibration	94
3.9	Light Intensity Models for Observed Objects	96
3.9.1	Simulation of Stars	96
3.9.2	Simulation of Planets and Moons	99
3.9.2.1	Step 1: Find Pixels Illuminated by Planet/Moon	99
3.9.2.2	Step 2: Find the Intensity of Reflected Light	101
3.9.2.3	Step 3: Find Number of Photoelectrons Generated at Each Illuminated Pixel	108
3.10	Intentional Defocusing	112
3.10.1	Star Centroiding	112
3.10.2	Determining Optimal Amount of Defocusing	113
3.11	Creating a Simulated Image	115
3.12	Measurement Covariance Matrix for Optical Line-of-Sight Observations	118
3.12.1	Classic Approach for Line-of-Sight Unit Vector Covariance	118
3.12.2	Line-of-Sight Unit Vector Covariance from Camera Parameters	121
3.12.3	Comparison of New Covariance Model with Results from Literature	125
3.13	Optical Sensor Model Case Study: Defocused Star Tracker	129
3.14	Optical Sensor Model Case Study: The MESSENGER Flybys	131

Chapter 4. Image Processing and Generating Optical Navigation Measurements 136

4.1	Summary of Possible Optical Navigation Measurements	137
4.1.1	Apparent Diameter and Centroid	137
4.1.2	Angle Between Horizon and Reference Star	139
4.1.3	Angle Between Surface Feature and Reference Star	141
4.1.4	Time of Star Occultation	143
4.2	Basic Image Processing	147

4.2.1	Linear Filters and Templates	147
4.2.2	Thresholding	150
4.2.3	Dilation and Erosion	152
4.2.4	Connected Components	153
4.3	Computing Distance to an Ellipse	155
4.3.1	Representations of an Ellipse	156
4.3.1.1	Implicit Equation for Ellipse	156
4.3.1.2	Standard Ellipse Parameters	157
4.3.1.3	Matrix Representation	158
4.3.2	Geometric Distance	159
4.3.3	Algebraic Distance	162
4.4	Ellipse Fitting	163
4.4.1	Basic Ellipse Fitting with Direct Least Squares Estimation	164
4.4.2	Improved Ellipse Fitting with Direct Least Squares Estimation	166
4.5	Robust Model Fitting	170
4.5.1	RANdom SAMple Consensus (RANSAC)	171
4.5.2	M-Estimator SAMple Consensus (MSAC)	173
4.6	Identifying the Location of Candidate Stars	175
4.7	New Algorithm for Planet Detection and Optical Navigation Measurements	177
4.7.1	Step 1: Rotate for Proper Sun Orientation	178
4.7.2	Step 2: Autonomous Planet/Moon Finding and Registration	181
4.7.3	Step 3: Extraction of Navigation Measurements	186
4.7.3.1	Spacecraft Position from Centroid and Apparent Diameter	186
4.7.3.2	Angle Between Horizon and Star	192
4.8	Comments on Image Exposure Time	194
4.9	Optical Navigation Measurement Covariance	198
4.9.1	Spacecraft Position from Centroid and Apparent Diameter	198
4.9.2	Angle Between Horizon and Star	201
4.10	Optical Navigation Measurement Sensitivity Matrices	204
4.10.1	Spacecraft Position from Centroid and Apparent Diameter	205
4.10.2	Angle Between Horizon and Star	205

Chapter 5. Optical Attitude Estimation	208
5.1 Existing Attitude Filtering Algorithms	209
5.2 Attitude Dynamics	213
5.3 A Review of the Classical Wahba Problem	214
5.4 Attitude Error and Covariance Relations	221
5.4.1 The Attitude Error Quaternion	221
5.4.2 Important Attitude Covariance Relations	222
5.4.3 The Wahba Problem Objective Function in Terms of Fisher Information	224
5.5 Development of SOAR Filter	226
5.5.1 Maximum Likelihood State Estimation	226
5.5.2 Discussion of SOAR Filter State Vector	231
5.5.3 Reformulation of SOAR Objective Function	231
5.5.4 Finding the Optimal State Update in SOAR	235
5.5.5 Finding the Optimal Covariance Update in SOAR	238
5.5.6 Propagation of State and Covariance for SOAR	241
5.6 Construction of SOAR Algorithm	243
5.7 Comparison of SOAR with Other Attitude Filters	244
5.7.1 Comparison of the SOAR Filter with the Multiplicative Extended Kalman Filter	244
5.7.2 Comparison of SOAR with Filter QUEST and REQUEST	246
5.7.3 Comparison of SOAR with Optimal-REQUEST	249
5.7.4 Comparison of the SOAR Filter with Extended-QUEST	251
5.8 Example: Estimation of Attitude and Gyro Bias	254
5.8.1 Theoretical Set-up for Example	254
5.8.2 Example Results	257
Chapter 6. Spacecraft Optical Navigation	262
6.1 Review of the Extended Kalman Filter (EKF)	263
6.2 Translational Spacecraft Dynamics Model	265
6.3 Jet Propulsion Laboratory Ephemeris Files	267
6.4 Case Study 1: Optical Navigation of a Robotic Spacecraft dur- ing a Venus Fly-by	267

6.5	Case Study 2: Human Cislunar Navigation	278
6.5.1	Attitude Estimation	280
6.5.2	Position and Velocity Estimation	281
Chapter 7.	Technology Development	294
7.1	Technology Readiness Level	295
7.2	Advancement Degree of Difficulty	297
7.2.1	Image Processing Algorithm	298
7.2.2	SOAR Filter	304
Chapter 8.	Conclusions	310
8.1	Conclusions	310
8.2	Future Work	313
Appendices		316
Appendix A.	Quaternion Mathematics	317
A.1	Definition of the Quaternion	317
A.2	Quaternion Multiplication	318
A.3	Complex Conjugate and Inverse of a Quaternion	319
A.4	Norm of a Quaternion	319
Appendix B.	Star Catalogs	321
Appendix C.	Numerical Algorithm for Computing the Direction to the Center of an Object from a Projected Ellipse	323
Appendix D.	Derivation of The QUaternion ESTimation (QUEST) Algorithm	326
Appendix E.	Derivation of Relation Between Fisher Information Matrix and Attitude Profile Matrix	332
E.1	Derivation of $\mathcal{F}_{\theta\theta}$ as function of \mathbf{B}	332
E.2	Derivation of \mathbf{B} as function of $\mathcal{F}_{\theta\theta}$	335

Bibliography	337
Index	366
Vita	369

List of Tables

2.1	Number of space sextant measurements taken during Experiment D-009 on Gemini IV and Gemini VII.	17
2.2	Comparison between Apollo 15 on-board (autonomous) and ground-based estimate of entry flight path angle.	22
3.1	Summary of performance specifications for the Mars Reconnaissance Orbiter's optical navigation camera.	62
3.2	Average bond albedo for selected celestial bodies.	105
3.3	Summary of performance specifications for Cypress Semiconductor STAR1000 radiation hard CMOS image sensor.	131
3.4	Summary of performance specifications for the MESSENGER spacecraft's Narrow Angle Camera.	134
5.1	Summary of noise levels for SOAR Filter validations (scenario 1).	258
5.2	Summary of noise levels for SOAR Filter validations (scenario 2).	260
6.1	Summary of MESSENGER images used in Venus fly-by case study. All images were taken with the MDIS wide angle camera.	271
6.2	Summary of image processing run times required to generate optical navigation measurements.	272
6.3	Initial conditions for MESSENGER fly-by of Venus at 21:18:29.737 UTC on 5 June 2007. These quantities are expressed in a Venus-relative reference frame.	274
6.4	Summary of spacecraft states on lunar return trajectory after each of the three TEI maneuvers.	280
7.1	Summary of recommended steps for maturing the optical navigation imaging processing algorithm from its current state (TRL 3) through TRL 6.	299
7.2	Summary of recommended steps for maturing the SOAR Filter from its current state (TRL 3) through TRL 6.	304

B.1 Summary of some of the star catalogs that are commonly used as a starting point for creating mission-specific star catalogs on modern spacecraft. 322

List of Figures

1.1	Potential navigation solutions for inertial state updates.	10
2.1	Space sextant used for autonomous navigation experiments in the Gemini program.	18
2.2	Star occultation photometer output for observation of Vega (brightest star in constellation Lyra) on Gemini X.	19
2.3	Schematic of the sextant (left) and scanning telescope (right) used for autonomous navigation in the Apollo Command Module.	21
2.4	Schematic of the space stadimeter operation.	24
2.5	OPNAV image of Deimos and background stars taken by Mariner 9 during Mars approach.	26
2.6	OPNAV image of Mars and background stars taken by Mariner 9.	27
2.7	Neptune system and background star field viewed by Voyager on approach.	29
2.8	Cumulative distribution of Cassini OPNAV images.	30
2.9	Voyager limb-scanning in the image plane.	32
2.10	The four single-frame mosaic OPNAV pictures for the Galileo encounter with the asteroid 951 Gaspra.	34
2.11	Centerfinding process used in the Galileo mission for a single node in a single-frame mosaic image.	36
2.12	Galileo OPNAV image of Ganymede and reference star taken on 3 June 1996.	39
2.13	Example OPNAV image of Mercury taken by MESSENGER on 10 January 2008.	46
2.14	Mars Reconnaissance Orbiter OPNAV image of Deimos and reference stars taken on 6 March 2006.	48
3.1	Graphical depiction of the relationship between the I , I' , B , and C frames.	55
3.2	Detector coordinate frames.	56
3.3	Diagram thin lens geometry.	59

3.4	Diagram of light rays from two objects at infinity through a simple thin lens system.	60
3.5	Optical layout for the Mars Reconnaissance Orbiter's optical navigation camera.	61
3.6	Line-of-sight geometry for a simple camera model.	63
3.7	Camera model with frontal image plane.	65
3.8	Notional diagram for the process of transferring electrons between adjacent pixels in a CCD array.	67
3.9	Examples of CCD blooming.	69
3.10	Graphical depiction of parallax.	74
3.11	Examples of baffle design for the Clementine star tracker and the Mars Reconnaissance Orbiter's Optical Navigation Camera.	78
3.12	Geometry of camera detector plane misalignment.	84
3.13	Effect of radial distortion on image.	86
3.14	Background noise on a dark frame (contrast significantly increased to make noise more visible).	89
3.15	Geometry of image of planet or moon.	101
3.16	Geometry of bidirectional reflectance.	103
3.17	Relationship between average single-scattering albedo and average bond albedo.	104
3.18	Geometry between source and observer.	109
3.19	Geometry for a camera system viewing a section of a planet. Image not to scale and angles are exaggerated for clarity.	111
3.20	Defocusing an image can enable subpixel accuracy through centroiding. Green 'x' shows true star location and red circle shows centroid estimate.	114
3.21	RMS centroid error along x-axis (contours in units of pixels) as a function signal-to-noise ratio and blur standard deviation.	116
3.22	Graphical depiction of changing error ellipses on the detector plane for two different line-of-sight unit vector observations with the same amount of angular error.	130
3.23	Simulated raw image data for star tracker image containing constellation Lyra.	132
3.24	Thresholded star tracker image containing constellation Lyra. Major stars in the constellation are labeled for clarity.	133
3.25	Optical layout for the MESSENGER spacecraft's Narrow Angle Camera.	134

3.26	Real and simulated OPNAV image from the MESSENGER fly-by of Venus on 7 June 2007.	135
3.27	Real and simulated OPNAV image from the MESSENGER fly-by of Mercury on 15 January 2008.	135
4.1	Geometry for apparent diameter and centroid measurement.	138
4.2	Geometry for star-horizon measurement.	140
4.3	Geometry for angular measurement between surface feature and reference star.	141
4.4	Geometry for star occultation.	144
4.5	Three-dimensional geometry for star occultation.	145
4.6	Geometry for star occultation as seen in the spacecraft orbit plane.	146
4.7	Three dimensional visualization of important kernel shapes.	149
4.8	Raw and thresholded images of Earth. The raw image was taken by the MESSENGER wide angle camera on 2 August 2005 during the spacecraft's Earth flyby.	151
4.9	Raw and thresholded images of Mercury. The raw image was taken by the MESSENGER wide angle camera on 14 January 2008 during the spacecraft's Mercury flyby.	152
4.10	Dilation and erosion of an OPNAV image of Earth.	154
4.11	Graphical explanation of a pixel that is (a) four-connected and (b) eight-connected.	155
4.12	Connected components analysis of an OPNAV image of Earth.	156
4.13	Graphical example of the ellipse points generated by the real roots of Eq. 4.40. One point represents the closest point on the ellipse (green) and the other point represents the furthest point on the ellipse (red). Example ellipse is centered at the origin with $a = 2$, $b = 1$, and $\phi = 25$ degrees.	163
4.14	Comparison of (a) geometric distance and (b) algebraic distance from an example ellipse (red) that is centered at the origin with $a = 2$, $b = 1$, and $\phi = 25$ degrees.	164
4.15	Example of RANSAC algorithm applied to noisy data along a line with a large number of outliers.	173
4.16	Example result of star finding algorithm on a raw image taken by the Clementine spacecraft's Star Tracker B on 20 March 2004.	176
4.17	Illumination direction.	179

4.18	Image rotation geometry.	180
4.19	Planet finding algorithm applied to example image containing Mercury taken by MESSENGER spacecraft's Narrow Angle Camera on 15 January 2008.	184
4.20	Planet finding algorithm applied to example image containing Venus taken by MESSENGER spacecraft's Narrow Angle Camera on 6 June 2007.	184
4.21	Planet finding algorithm applied to example image containing Earth taken by MESSENGER spacecraft's Narrow Angle Camera on 2 August 2005.	184
4.22	Planet finding algorithm applied to example image containing Earth taken by MESSENGER spacecraft's Narrow Angle Camera on 2 August 2005.	185
4.23	Planet finding algorithm applied to example image containing Phobos taken on 22 August 2004 by the Mars Orbiter Camera (MOC) on ESA's Mars Express spacecraft.	185
4.24	Planet finding algorithm applied to example image containing lunar horizon taken by the Clementine spacecraft's Star Tracker B on 20 March 2004.	186
4.25	Error ellipse for instantaneous position fix from centroid and apparent diameter, with and without minimization of light profile residuals. Results for simulated image of Moon at a range of 22,433 km with a $10.5^\circ \times 10.5^\circ$ FOV camera.	193
4.26	Example star-horizon measurements from image taken by the Clementine Star Tracker B on 20 March 2004.	195
4.27	Error ellipse for instantaneous position fix from centroid and apparent diameter. Results for simulated image of Moon at a range of 200,636 km with a $1.5^\circ \times 1.5^\circ$ FOV camera.	202
4.28	Geometry of star-horizon measurement for a spherical planet or moon.	206
5.1	SOAR Filter flow chart.	245
5.2	Estimated attitude errors for the SOAR Filter in scenario 1 (normal case with small errors).	258
5.3	Estimated bias errors for the SOAR Filter in scenario 1 (normal case with small errors).	259
5.4	Comparison of the estimated attitude errors for the MEKF, Filter QUEST, and the SOAR Filter in in scenario 2 (stressing case with large errors).	260

5.5	Comparison of the estimated bias errors for the MEKF, Filter QUEST, and the SOAR Filter in in scenario 2 (stressing case with large errors).	261
5.6	Zoom-in of roll-axis attitude error for scenario 2 (stressing case with large errors).	261
6.1	Image geometry for June 2007 MESSENGER fly-by of Venus.	270
6.2	Measurement error from instantaneous position fix generated from apparent diameter and centroid of Venus.	273
6.3	Spacecraft position covariance at the end of the Venus fly-by.	274
6.4	Position error for June 2007 MESSENGER Venus fly-by.	276
6.5	Velocity error for June 2007 MESSENGER Venus fly-by.	276
6.6	Attitude error from SOAR Filter.	282
6.7	Gyro bias error from SOAR Filter.	282
6.8	Image geometry for 19-24 February 2024 lunar return trajectory with a three-burn TEI sequence.	283
6.9	Synthetic images of the Earth and Moon (with no surface texture map) at point <i>a</i> (ID 1) on the reference lunar return trajectory.	285
6.10	Synthetic images of the Earth and Moon (with no surface texture map) at point <i>b</i> (ID 22) on the reference lunar return trajectory.	285
6.11	Measurement error from instantaneous position fix generated from apparent diameter and centroid of the Earth.	286
6.12	Measurement error from instantaneous position fix generated from apparent diameter and centroid of the Moon.	286
6.13	Filtered position error for autonomous lunar return using only apparent diameter and centroid measurements of the Earth and Moon.	289
6.14	Filtered velocity error for autonomous lunar return using only apparent diameter and centroid measurements of the Earth and Moon.	289
6.15	Filtered position error for autonomous lunar return using only star-horizon measurements measured from the Earth and Moon.	290
6.16	Filtered velocity error for autonomous lunar return using only star-horizon measurements measured from the Earth and Moon.	290
6.17	Filtered position error for autonomous lunar return using apparent diameter, centroid, and star-horizon measurements.	291

6.18	Filtered velocity error for autonomous lunar return using apparent diameter, centroid, and star-horizon measurements. . .	291
6.19	Entry FPA error generated by propagating current estimate state and covariance forward to entry interface. Results are for filter that only uses apparent diameter and centroid measurements of the Earth and Moon.	292
6.20	Entry FPA error generated by propagating current estimate state and covariance forward to entry interface. Results are for filter that only uses star-horizon measurements measured from the Earth and Moon.	293
6.21	Entry FPA error generated by propagating current estimate state and covariance forward to entry interface. Results are for filter that uses apparent diameter and centroid measurements of the Earth only, and star-horizon measurements from the Moon only.	293
7.1	The Technology Readiness Level (TRL) scale.	296
7.2	Notional schematic of major GNC subsystem interfaces that are important for demonstration of a relevant environment for the image processing algorithm and SOAR Filter.	302
7.3	Image of author deploying a three degree-of-freedom attitude control experiment on a parabolic microgravity flight in April 2003.	306
7.4	Images of PARADIGM deployment taken from the Space Shuttle Endeavor (STS-127) on 30 July 2009.	308
C.1	Graphical depiction of the perspective projection of a sphere onto the detector plane.	324
D.1	Flow chart for QUEST algorithm.	331

Chapter 1

Introduction

Recent years have seen ambitious robotic exploration missions to other planets and a renewed interest in sending humans beyond low Earth orbit (LEO). Developments in the fields of both robotic spaceflight and human spaceflight are creating increased pressure for advancements in autonomous spacecraft operation. A particularly challenging problem is spacecraft navigation independent of any Earth-based tracking or updates. For robotic missions, autonomous navigation may be necessary for high-speed fly-bys of a planet or for fly-bys that occur when the Sun is between the Earth and the planet. Autonomous navigation may also help reduce the tracking demands on ground based infrastructure. For crewed missions, autonomous navigation would enable the crew to safely navigate the spacecraft back to Earth in the event of a communication system failure.

The objective of the present research is to investigate navigation solutions that would enable a spacecraft to autonomously navigate in a planetary system. Optical navigation techniques are proposed as a solution to this problem. Specifically, the focus is on navigation in planetary space, not in interplanetary space. Examples of operations in planetary space include, but

are not limited to, a spacecraft in orbit about a planet/moon, a spacecraft executing a fly-by of a planet/moon, and a spacecraft in a cislunar transfer orbit. Some examples of operations in interplanetary space, where the techniques investigated in this dissertation are not generally valid, include a spacecraft in a heliocentric cruise phase and a spacecraft executing a fast fly-by of a small solar system body (something other than a planet or moon, e.g. comets, asteroids, etc.). Throughout the following discussions, the phrase “autonomous navigation” refers to navigation without contact with Earth.

1.1 Historical Perspective and Motivation

1.1.1 Interplanetary Robotic Spaceflight

Interplanetary robotic missions have historically relied on a combination of radiometric tracking and optical observations for navigation. Radiometric tracking data, usually provided by the Deep Space Network (DSN), are the primary means of navigation for most flight regimes. DSN is a network of large antennas (34 m diameter and 70 m diameter) operated by the Jet Propulsion Laboratory (JPL), with deep space communication facilities located in California (USA), Spain, and Australia. More background information on DSN may be found in [1], [2], and [3]. Almost all lunar missions and numerous interplanetary missions have relied entirely on radiometric tracking (from DSN or similar assets) for external position and velocity updates. During the planetary approach phase, some spacecraft have also relied on optical observations to improve the estimate of the spacecraft state relative to the tar-

get planet or moon. With a few exceptions, to be discussed more in Chapter 2, these optical observations were processed on the ground. The radiometric tracking data and optical data were combined on the ground and an updated state estimate was uplinked to the spacecraft.

Recent robotic missions such as New Horizons, which was launched in January 2006 and is now in transit to Pluto, are beginning to perform more complex tasks while operating at greater distances from the Earth. Operating at such large distances from the Earth requires increased autonomy due to the time delay associated with sending navigation data from a distant planet, processing that data on the ground, and uplinking a new state estimate to the spacecraft. This time lag may be too great for some applications.

Further, the number of lunar and planetary spacecraft simultaneously operating continues to grow. Current projections indicate that the Earth-based resources used to track and communicate with these spacecraft face significant challenges in keeping up with the number of required links as well as the total required data rate.[4, 5] To respond to these growing needs and aging infrastructure, the DSN program initiated an antenna modernization effort in February 2010 that is expected to last through 2025.[6] Although DSN will likely remain the primary system for deep space communication and radiometric tracking, there have been some recent additions to our ability to track and communicate with lunar and planetary spacecraft. Notable among these additions is a Ka-band antenna network opened in 2007 and operated by the National Aeronautics and Space Administration (NASA) at the White Sands

Test Facility in New Mexico.[7] A second notable example is the Indian Deep Space Network (IDSN) in Byalalu, India, which was opened in 2008 to track and communicate with Chandrayaan-1, the country's first lunar satellite.[8]

Increased spacecraft autonomy will help reduce the demands and constraints on DSN and other Earth-based radiometric tracking facilities. The increased flexibility that results from this autonomy may help bridge the projected capability gap in planetary communication and tracking.

1.1.2 Human Spaceflight

The discussion that follows contains a few brief comments on the post-Apollo developments in human spaceflight beyond LEO, along with a description of the current events that are shaping the direction of human spaceflight today.

Human spaceflight has been limited to LEO since the last Apollo mission, Apollo 17, returned from the moon on December 19, 1972.[9] After many years in LEO with the Apollo-Soyuz, Skylab, and Space Shuttle programs, President George H.W. Bush announced the Space Exploration Initiative (SEI) in 1989. This program outlined a 30-year plan for an orbital space station, a permanent base on the Moon, and a human mission to Mars.[10–12] The plans that were developed as part of SEI never happened due to political pressure that resulted from a difficult roll-out and a large cost estimate. Human space exploration beyond LEO was officially removed from the U.S. national agenda by the Clinton Administration in 1996.[13] Throughout this time the

Space Shuttle and the development of the International Space Station (ISS) remained the cornerstones of U.S. human spaceflight.

During the late 1990s and early 2000s, there were numerous efforts to develop a vehicle to replace the Space Shuttle for delivering/returning crew and cargo to/from the ISS. Most notable among these attempts were the X-38 Crew Return Vehicle (CRV) and the Orbital Space Plane (OSP). The programs for all of these LEO vehicles were canceled before the systems became operational. During this time period there were also numerous paper studies on human exploration of the Moon and Mars (e.g. Exploration Blueprint, Mars Design Reference Missions).[14, 15] Then, after the Space Shuttle Columbia accident on February 1, 2003,[16] President George W. Bush introduced the Vision for Space Exploration (VSE) on January 14, 2004.[17] The VSE called for the United States to return to the Moon by 2020 on its way to exploring destinations beyond Earth's neighborhood. In response to the VSE, NASA created the Constellation Program to develop the systems necessary to complete this vision. To shift focus and resources to this new program, NASA also announced its decision to end the Space Shuttle program in 2010 after nearly 30 years of operation. After approximately 5 years of responding to President Bush's VSE, NASA's direction and objectives were reassessed after a presidential change of power. In June 2009, the Obama Administration created the Review of U.S. Human Spaceflight Plans Committee to review the entirety of U.S. human space flight, including the Space Shuttle, ISS, and Constellation Program. The committee's report, "*Seeking a Human Spaceflight Program*

Worthy of a Great Nation,” was delivered in October 2009.[18] This report concluded that the Constellation Program was underfunded and that NASA’s current objectives were not appropriately matched with the agency’s projected funding profile. At the time that this dissertation was written, the future of the Constellation Program and, therefore, the future direction of U.S. human spaceflight beyond LEO was uncertain.

Regardless of the future of the Constellation Program, the recurring interest in human exploration beyond LEO is taken as evidence of an underlying national interest in human space exploration. The primary obstacles for these human space exploration programs continue to be a constrained budget, a large risk to the crew, and a very long development period. These challenges create difficulty in sustaining sufficient support for human space exploration programs across multiple presidential administrations and congressional sessions. Therefore, from an engineering standpoint, it is critical to develop technologies and capabilities that address these issues by reducing cost, reducing the risk to the crew, and collapsing the schedule.

Increasing vehicle autonomy is a widely cited mechanism for achieving all of these objectives. One of the major contributing factors to the cost of the Space Shuttle and ISS programs is the large number of people it takes to operate these systems. The Space Shuttle Program, for example, had a standing work force of over 12,500 individuals in the year 2009.[18] If increasing vehicle autonomy can greatly reduce the work force requirements for mission control of crewed vehicles, then significant cost savings could be realized. Ad-

ditionally, increased vehicle autonomy (or the ability to operate autonomously if necessary) may increase vehicle reliability if properly implemented. A communication system failure, for example, may require a crewed vehicle to autonomously return to Earth. Finally, development of technologies that enable autonomous human spacecraft operation today will reduce the time required to field these technologies in future systems.

1.2 Preliminary Navigation Considerations

During operations in LEO, the data required for navigation may be obtained through traditional spacecraft navigation methods, such as an onboard Global Positioning System (GPS) receiver or radiometric tracking. As the distance between the spacecraft and Earth increases, some of the methods used in LEO become problematic due to design (e.g. GPS signals are designed to transmit towards the Earth) and/or poor geometry. To address this difficulty, previous spacecraft operating in planetary space have relied primarily on a combination of onboard inertial measurements and inertial state updates sent from the ground. These ground updates are commonly generated from radiometric tracking. Optical images taken by the spacecraft, although processed on the ground, have also been widely used to supplement the radiometric tracking data during an encounter with a planet, moon, or asteroid. Traditionally, an Inertial Measurement Unit (IMU) is used to propagate the state (dead reckoning) between inertial state updates provided from an external source. For many spacecraft applications, this external source is a state

estimate uplinked from the ground. An IMU is a single device that contains both accelerometers and gyros. Between the external state updates, the IMU generated solution will drift. The difficulty here lies primarily with the accelerometers and the associated estimate of the spacecraft position; the gyros may be inertially updated using on-board star trackers.

Autonomous navigation, therefore, requires some form of external information to augment the measurements from the IMU. The following two sections, therefore, discuss some introductory notions related to (1) inertial navigation and (2) sources of autonomous external navigation updates.

1.3 Inertial Navigation

Inertial navigation represents one extreme in the spectrum of autonomous navigation. Inertial navigation allows the spacecraft to track its position, velocity, attitude, and attitude rate as a function of time without requiring measurements from the outside environment (e.g. radiometric tracking, observations of stars, etc.). Inertial changes in attitude are easily measured using the IMU's gyros. Detecting changes in the translational states is more difficult. Because accelerometers do not measure the acceleration due to gravity, propagation of the spacecraft position and velocity is dependent on a gravity model stored in the spacecraft computer. When non-gravitational forces are applied to the vehicle (e.g. propulsive maneuver, waste venting, etc.), the total acceleration may be computed by combining the predicted gravitational acceleration from the gravity model with the sensed accelerations measured by

the accelerometers. The spacecraft position and velocity may be tracked over time by integration of these accelerations from some set of initial conditions.

Unfortunately, knowledge of the gravity field is imperfect and the gravity models used for real-time applications require approximations that may introduce significant errors for long propagation periods (e.g. truncation of the spherical harmonic representation of the planet and/or moon gravity fields). Further, there are numerous sources of error in the acceleration measured by the accelerometer. Due to these errors, the solution achieved by simply integrating the measured/predicted accelerations accumulates error over time and the system requires external updates to maintain an accurate estimate of the spacecraft position and velocity.

1.4 Sources of Autonomous External Navigation Updates

Determining the degree of autonomy that should be incorporated into a spacecraft is an important design decision. Autonomous attitude determination may be performed by imaging a star field and matching the observed star patterns to known patterns from a star catalog. This process is well understood and is a function performed by many commercially available star trackers.[19] Autonomous updates of the spacecraft translational states are significantly more difficult.

The desire for autonomous lunar and planetary navigation is a long standing issue that has led to numerous studies since the 1960s. Autonomous

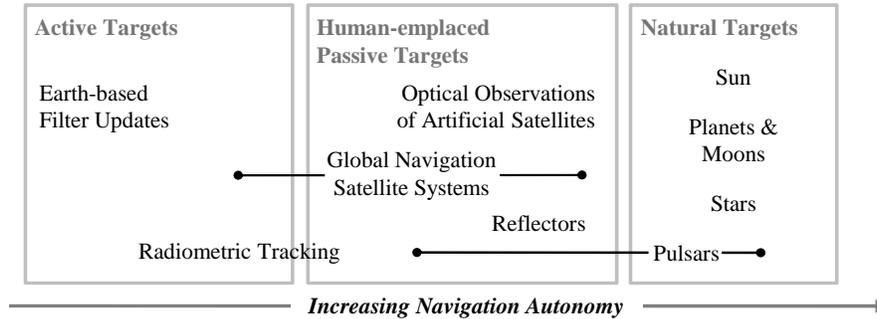


Figure 1.1: Potential navigation solutions for inertial state updates.

navigation techniques (mostly optical) have been tested and widely used during the planetary approach phase of robotic exploration missions. Numerous examples of these robotic missions are discussed in detail in Chapter 2. Because the required degree of autonomy will vary from mission to mission, a spectrum of navigation solutions must be considered (see Fig. 1.1).

The approaches presented towards the right half of Fig. 1.1 represent solutions that would allow for autonomous inertial navigation updates. These solutions rely primarily on observations of natural targets. The resulting measurements typically consist of optically acquired images of stars and a nearby celestial object or the time of arrival of pulses from well-known pulsars. Although the theory and algorithms for pulsar-based navigation is relatively mature,[20] there are difficulties in the density of well-modeled pulsars and the size/sensitivity of the required X-ray detector. Pulsars are shown to span “Natural Targets” and “Human-emplaced Passive Targets” in Fig. 1.1 because some pulsar-based navigation schemes require a station at a known location to monitor the pulsar signal and then transmit timing information about the

observed signal to other spacecraft. These other spacecraft may then perform relative navigation with respect to this reference station using the pulsar signal. Other pulsar-based navigation schemes are completely autonomous.

Therefore, optical navigation (OPNAV) techniques are proposed as a solution to the problem of navigating in a planetary system without requiring navigation information from Earth. In such a scenario, the spacecraft would collect optical images of naturally occurring objects. Then, by coupling accurate angular measurements with precise target position information, the resulting data may be used to produce an estimate of the spacecraft's inertial position. The use of natural targets allows for inertial state updates without requiring the emplacement of additional space-based assets and without the need for contact or communication with the Earth. Use of natural targets also helps to create a robust navigation architecture. The observation of man-made targets may introduce additional maintenance, infrastructure, overhead, and failure modes not seen with natural targets. It should also be noted that, despite these advantages, natural targets rarely have the precision associated with man-made systems specifically designed for navigation.

1.5 Contributions

This work provides a detailed discussion of the tools and techniques required to create an autonomous optical navigation system for a spacecraft operating in the vicinity of a planet or moon. Although substantial work has been performed in this area since the 1960s, this dissertation introduces

numerous advancements to the current state-of-the-art. The sections that follow discuss each of the major contributions in more detail.

1.5.1 Optical Sensor Model

A new optical sensor model is developed. Individual error sources (e.g. alignment, optics, detector noise, etc.) are discussed from the perspective of generating optical navigation measurements. These observations are used to create synthetic images of stars and planets that can be used to test optical navigation image processing algorithms. This new model also allows for direct construction of the covariance matrix for a line-of-sight unit vector observation as a function of camera parameters. The new covariance model is shown to reduce to various simpler forms of the covariance found in the literature under the appropriate assumptions. This contribution is the subject of Chapter 3.

1.5.2 Image Processing

A new application of image processing techniques is presented that allows for the autonomous extraction navigation information from a raw image. The performance of this image processing algorithm is demonstrated on (1) real images from previous spacecraft missions and (2) synthetic images generated using the newly developed optical sensor model (contribution 1). Analytic expressions for the measurement covariance matrices and measurement sensitivity matrices are developed for each of the optical navigation measurement types discussed. This contribution is the subject of Chapter 4.

1.5.3 The Sequential Optimal Attitude Recursion (SOAR) Filter

A new nonlinear attitude filter called the Sequential Optimal Attitude Recursion (SOAR) Filter is developed. The complete derivation of the SOAR Filter is presented, starting from the perspective of maximum likelihood estimation. Detailed theoretical comparisons between the SOAR Filter and a number of existing attitude filters places this new attitude filter in the context of earlier work. Simulation results indicate that the SOAR Filter is capable of providing superior performance relative to existing attitude filters when angular errors are large. When the errors are small, the performance may be shown to be identical to the Multiplicative Extended Kalman Filter (MEKF). This contribution is the subject of Chapter 5.

1.5.4 End-to-end Optical Navigation Performance Assessment

The results of the first three contributions are combined to perform an end-to-end assessment of the performance of a completely autonomous optical navigation system in two different scenarios. In the first scenario, real images from the June 2007 MESSENGER fly-by of Venus are autonomously processed and the state is estimated throughout the fly-by using only these optical navigation measurements. In a second scenario, synthetic images of the Earth and Moon are generated for a spacecraft on a February 2024 lunar return trajectory. As with the first case study, these images are autonomously processed and the entire lunar return is performed using only measurements from the spacecraft IMU and camera. The results of this second case study

are put in the context of meeting a representative entry flight path angle requirement for a crewed vehicle on a lunar return. The contributions of these two case studies are subject of Chapter 6.

Chapter 2

Historical Background on Optical Navigation

2.1 Extension of Terrestrial Optical Navigation to Space Applications

Optical navigation has been used by explorers on Earth for centuries. The use of stars (mainly Polaris) and the Sun for navigation dates back to at least the 1400s. Early navigation consisted mostly of determining latitude by measuring the angle between Polaris and the horizon. Difficulties with accurate time keeping, however, prevented the determination of longitude until the development of Harrison's chronometer in the 1700s.[21] Although the accuracy of navigation instruments and star charts have improved, the fundamental techniques used by modern practitioners of celestial navigation (usually as a back-up to GPS and other man-made navigation aids) have changed little in the past 200 years.

Many of the fundamental measurement types used in terrestrial navigation may be extended for use in space applications. These measurements may be made manually (through a device such as a space sextant or a space stadimeter) or autonomously through an optical image of a planet/moon and a background starfield. Some of the common optical measurements (discussed in detail in Chapter 4) include: (1) planet/moon apparent diameter, (2) the angle

between a planet/moon horizon and a reference star, (3) the angle between a planet/moon surface feature and a reference star, and (4) the time at which a planet/moon occludes a reference star. The star-horizon measurements used for spacecraft navigation represent a direct parallel to the sextant observations used by seafarers on Earth since the 1700s.[22] The use of apparent diameter measurements, star-horizon measurements, and star-feature measurements for navigation in space applications has been well understood since at least the early 1960s.[23, 24] Although much of the early groundbreaking work in optical navigation was performed as part of the Gemini program and Apollo program, advancements in autonomous navigation and optical navigation by robotic spaceflight has far outpaced that seen in human spaceflight since the 1970s.

2.2 Optical Navigation in Human Spaceflight

Some of the first demonstrations of optical spacecraft navigation were performed in the human spaceflight programs of the 1960s. Most of these advancements were made in direct preparation for the Apollo missions to the moon. Since the Apollo program ended, however, there have been few additional advancements in optical navigation from the human spaceflight community. This is likely due to the fact that, with the exception of Apollo, human spaceflight has been limited to LEO, where the need for optical navigation is less pronounced.

2.2.1 Gemini

Numerous optical navigation experiments were performed as part of the Gemini program, many of which were in direct preparation for the Apollo program. These experiments were primarily coordinated by the Air Force Avionics Laboratory at Wright-Patterson Air Force Base, which had been developing optical navigation techniques since the early 1960s.

The first optical navigation experiment, Experiment D-009, was intended to demonstrate the use of optical observables for autonomous space navigation for the first time.[25] This experiment was flown on Gemini IV (June 1965) and Gemini VII (December 1965). Although experiment plans called for the testing of a space sextant (Fig. 2.1) and a space stadimeter, only the space sextant was actually used. The space stadimeter was built and qualified, but never flew on Gemini.[25] A summary of the measurements taken as part of this experiment is shown in Table 2.1. The experiment was highly successful and helped lay the foundation for the optical navigation system used in the Apollo program.

Table 2.1: Number of space sextant measurements taken during Experiment D-009 on Gemini IV and Gemini VII.[25]

Measurement Type	Gemini IV	Gemini VII
Earth horizon to star	45	37
Moon horizon to star/planet	-	5
Star to star	47	6
Single star	-	8

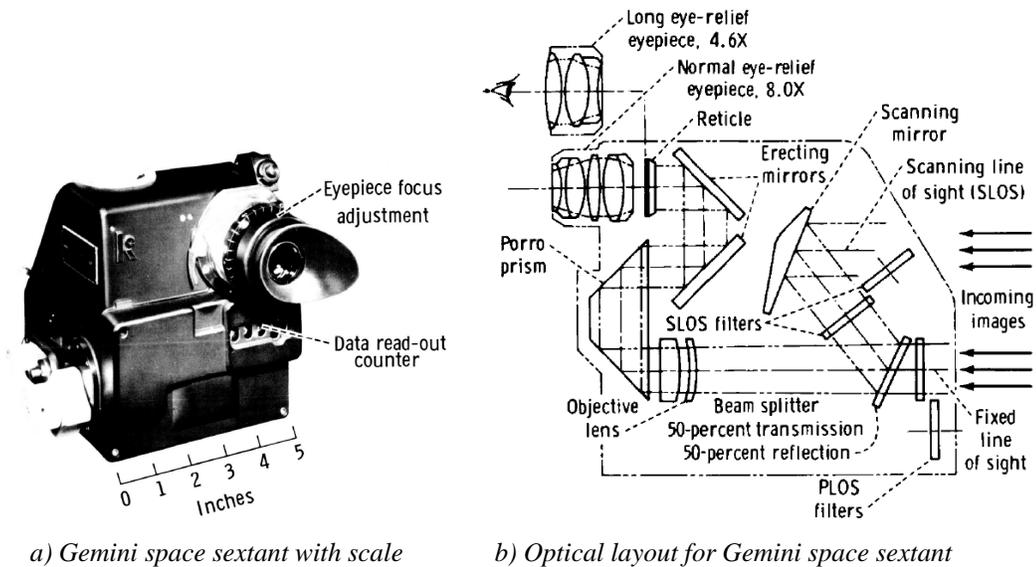


Figure 2.1: Space sextant used for autonomous navigation experiments in the Gemini program. Original figure from [26], courtesy of NASA.

The second experiment, Experiment D-005, was intended to demonstrate how star occultation measurements could be used for autonomous spacecraft navigation.[27] This experiment was flown on Gemini VII (December 1965) and Gemini X (July 1966). To determine the time of star occultation, the astronauts used an instrument called a star occultation photometer. This device was used to create the attenuation curve (observed light intensity relative to unattenuated intensity as a function of time) of an observed star as that star passed through the Earth’s atmosphere. A failure of the star occultation photometer on Gemini VII resulted in no occultation data being collected during this mission.

Numerous measurements were made with the star occultation photome-

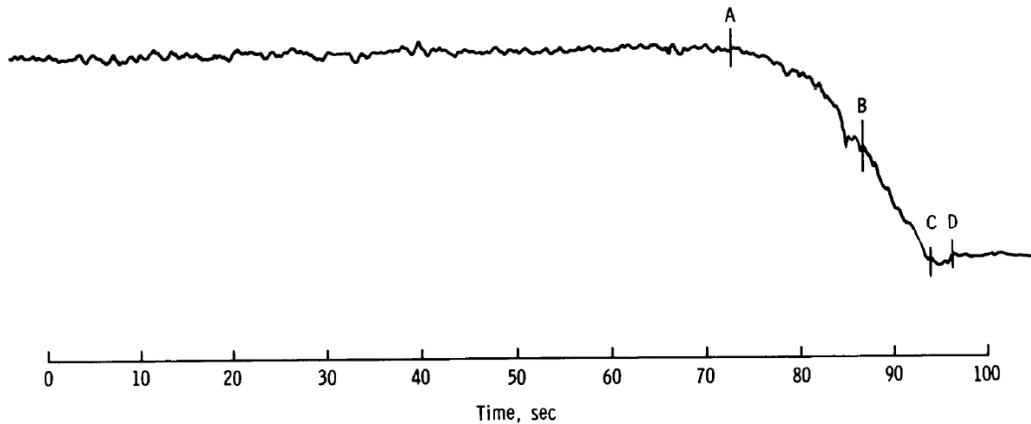


Figure 2.2: Star occultation photometer output for observation of Vega (brightest star in constellation Lyra) on Gemini X. The line-of-sight to the star began to pass through Earth’s upper atmosphere at point A and appeared to exhibit a linear decrease in intensity as it passed through the lower atmosphere between points B and C. Original figure from [27], courtesy of NASA.

ter on Gemini X. The astronauts reported difficulty tracking a star through the Earth’s airglow, where the star would momentarily disappear and then reappear before moving below the Earth’s dark horizon. This highlights some of the difficulties of using star occultation measurements on a planet with an atmosphere. An example output of a successful star occultation photometer measurement on Gemini X may be seen in Fig. 2.2.

The third experiment, T-002, flown on Gemini XII (November 1966) was intended to validate the ground based simulations for space sextant measurements.[26] The focus of this experiment was on the effects of the space environment on the performance of manual sightings taken by crew members. As part of this study, the astronauts took sightings with the Gemini space

sextant (Fig. 2.1) with and without the space suit helmet on. Results indicated that the ground based simulations accurately estimated performance of the instrumentation and crew in the actual space environment.[26]

2.2.2 Apollo

The Apollo Guidance, Navigation, and Control (GNC) system was originally designed to be capable of supporting a return to Earth completely independent of Earth-based resources. Unfortunately, limitations in on-board computer memory forced some functions to be shifted to ground tracking facilities and Mission Control.[28] Even after this reduction in capability, the crew maintained sufficient on-board navigational capability to return to Earth in the event of a communications system failure. Although the capability for autonomous navigation was retained as a backup, it was not regularly used because of extremely high demands on crew time.

The key instrument that enabled this on-board capability was a specially designed sextant used to measure the angle between a selected Earth or Moon feature and a reference star. This device, operated manually by one of the astronauts, was capable of making angular measurements with accuracy on the order of 10 arcsec (made possible through a 28 power eyepiece).[28] Unfortunately, the field of view for the high power eyepiece was so small that a wide field scanning telescope was also required. Schematics of the sextant and scanning telescope, shown in Fig. 2.3, illustrate how the angle between the Earth/Moon landmark and the reference star was actually measured.[29]

More detailed discussions of both the sextant and scanning telescope may be found in [29], [30], and [31].

Astronauts claim to have found the star-horizon measurement easier to obtain than the star-landmark measurement. In practice, the trunnion angle was manually adjusted by the astronaut until the image of the reference star was superimposed on the lunar horizon (similar to how a traditional sextant works for Earth applications). When a measurement was ready to be recorded, the astronaut could press a “mark” button to record the time and trunnion angle. This information was passed to the spacecraft computer where the Cislunar Navigation Program, called Program 23 (P-23), [32, 33] would deter-

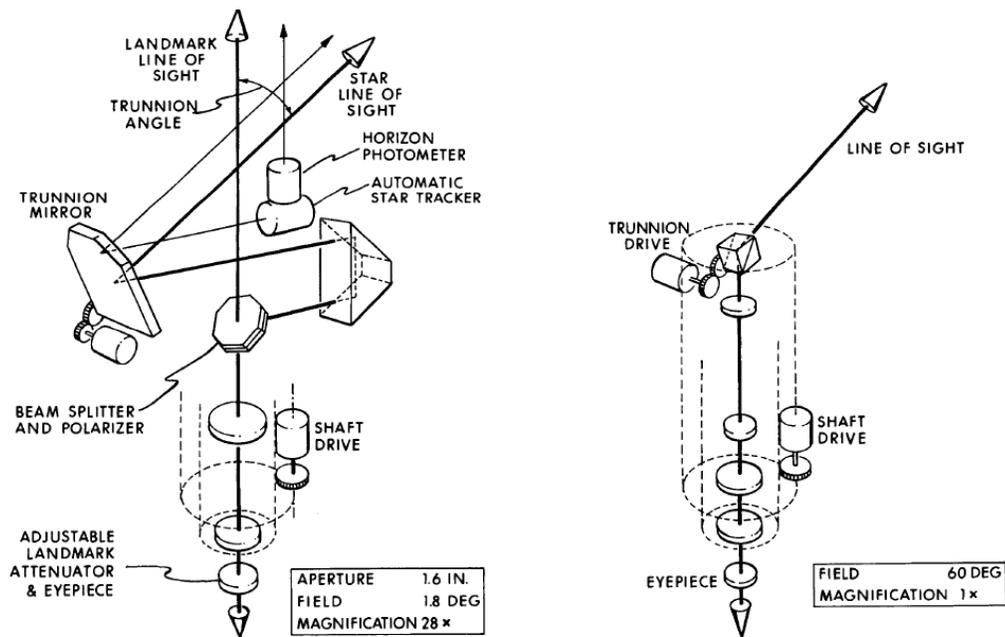


Figure 2.3: Schematic of the sextant (left) and scanning telescope (right) used for autonomous navigation in the Apollo Command Module. Original image from [29], courtesy of NASA.

Table 2.2: Comparison between Apollo 15 on-board (autonomous) and ground-based estimate of entry flight path angle.[32]

Time to Entry Interface	On-board Entry FPA	Ground-Based Entry FPA	Entry FPA Difference
-58.2 hr	-7.23 deg	-6.69 deg	-0.54 deg
-43.7 hr	-7.22 deg	-6.69 deg	-0.53 deg
-31.5 hr	-6.56 deg	-6.50 deg	-0.06 deg
-22.0 hr	-6.55 deg	-6.50 deg	-0.05 deg
-19.0 hr	-6.61 deg	-6.50 deg	-0.11 deg
-6.0 hr	-6.26 deg	-5.82 deg	-0.44 deg
-1.4 hr	-6.80 deg	-6.49 deg	-0.31 deg

mine the state update. If the state update was within the expected bounds, the measurement was permitted to update the state. A detailed discussion of P-23 algorithm is provided in [33].

The performance of this system was demonstrated on Apollo 8 when Jim Lovell took over 200 sextant sightings and was able to calculate the closest approach to the Moon (using only on-board resources) to within 2.5 km of that computed by post-flight analysis.[28] During the return, Lovell also demonstrated that this method provided sufficient accuracy to meet the reentry requirements.[28, 34] This capability was demonstrated again during later Apollo missions. Data from Apollo 15 (see Table 2.2)[32, 35] demonstrates that the on-board flight path angle (FPA) estimates were sufficient to meet the reentry requirements (error of ± 0.5 deg desired, ± 1.0 deg required). Note that the difference between the ground-based entry FPA estimates and on-board entry FPA estimates collapse to below the reentry requirement as time approaches entry interface (entry interface occurred 308.5 hours after liftoff

for Apollo 15).

2.2.3 Skylab

Skylab was launched in May 1973 and was the United States' first space station. Three different crews visited the station in 1973 and 1974. The Manual Space Navigation Experiment (DOD/NASA Skylab Experiment T-002) was performed in 1973 by the second and third Skylab crews.[36]

This experiment looked at the performance of a space sextant and a space stadimeter. Results of this experiment indicated that the top of Earth's airglow provided the best reference for star-horizon measurement. Determining the height of the observed airglow above the Earth's surface, however, introduced another unknown which had to be estimated. Additionally, the Skylab program provided an opportunity to test the US Air Force space stadimeter that was not able to be flight-tested during the Gemini program. The space stadimeter worked by taking three line-of-sight measurements of the Earth's horizon (see Fig. 2.4) to estimate the planet's apparent diameter, and therefore estimate the altitude of the Skylab space station. Detailed discussions on the operation of the space stadimeter may be found in [36], [37], and [38].

2.2.4 Space Shuttle

Unlike its immediate predecessors in US human spaceflight, the Space Shuttle was not designed with the capability for onboard optical navigation by the crew. A nice comparison of the Apollo and Space Shuttle navigation

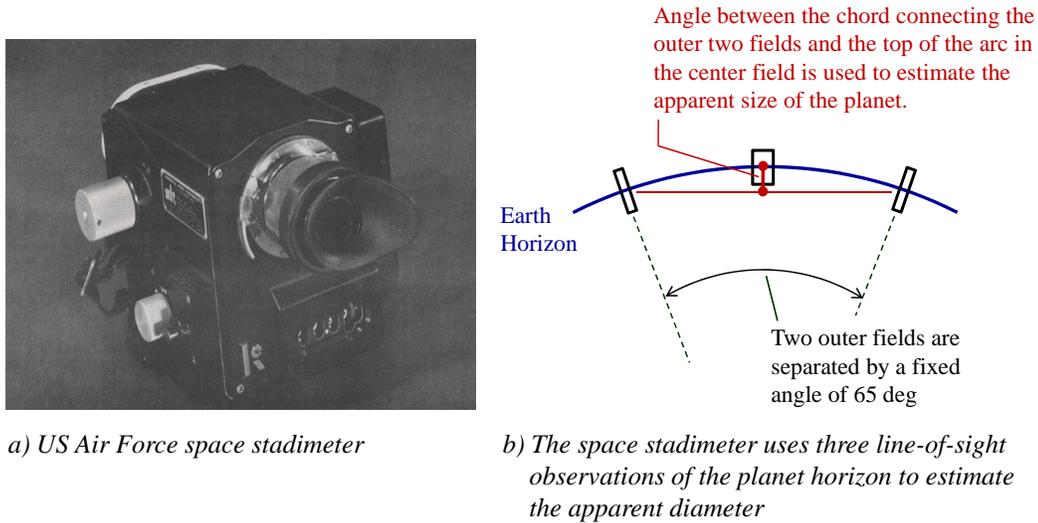


Figure 2.4: Schematic of the space stadimeter operation. Picture on left is from [38], courtesy of NASA.

systems may be found in [39].

The closest instrument the Space Shuttle has to the space sextant used on Gemini, Apollo, and Skylab is the Crewman Optical Alignment Sight (COAS). Like the space sextant, this is a manually operated device used to take star sightings. Unlike the space sextant, the COAS is used for attitude estimation. The COAS may be used to realign the IMU if the attitude error drifts beyond about 1.4 degrees. Beyond this point, the *a-priori* attitude error is too large for the star tracker to acquire and track stars. In this scenario, the attitude must be manually realigned to within 1.4 degrees before the star trackers can take over alignment process with more precision. The COAS is also used to track target vehicles.[39]

2.3 Optical Navigation in Robotic Spacecraft

Many planetary missions have used optical navigation. What follows is a brief summary of relevant past missions. Although the following discussion is not exhaustive (there are many other missions that have used optical navigation), it does provide a summary of key historical advancements and describes the current state of the art in optical navigation for robotic spacecraft.

2.3.1 The Mariner Missions to Mars

Many of the early missions to Mars relied solely on radiometric tracking from DSN. Mariner 4 (launched on November 28, 1964), Mariner 6 (launched on February 24, 1969), and Mariner 7 (launched on March 21, 1969) all used only radiometric tracking for navigation during all mission phases. All three of these missions performed Mars flybys.

Mariner 9 launched on May 30, 1971 and inserted into Mars orbit on November 14, 1971. Mariner 9 was the first spacecraft to enter into orbit about another planet. Although preliminary optical navigation experiments were flown on Mariner 6 and Mariner 7,[40] Mariner 9 was the first robotic mission to incorporate optical navigation measurements into the mission orbit determination process. The Mariner 9 spacecraft used optical images of Mars and its moons (Phobos and Deimos) along with background stars to improve navigation during the approach to Mars.[41–43] During this mission a total of 21 OPNAV images were taken (18 of Deimos and 3 of Phobos).[44] An example image of Deimos and background stars taken during the 1971 Mars approach

is shown in Fig. 2.5. A typical OPNAV image of Mars is shown in Fig. 2.6. The TV cameras used to create both of these images used vidicons with 700 scan lines and 832 pixels per line.

OPNAV images such as those shown in Fig. 2.5 and Fig. 2.6 were transmitted to Earth where they were combined with radiometric tracking data. The algorithm used to detect the location of the lit Mars limb was fairly simple. Each scan line was searched for a set of three pixels that were above a specified threshold. The first pixel in this set was identified as the limb location. The three-pixel width was chosen to prevent single-pixel noise spikes from generating incorrect limb locations.[43]

The success of optical navigation on Mariner 9 provided an early demonstration of the effectiveness of this technique and laid the foundation for the

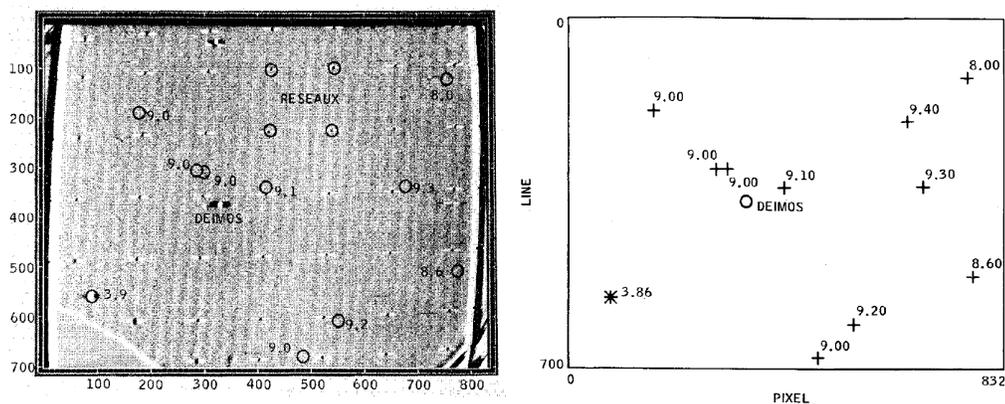


Figure 2.5: OPNAV image of Deimos and background stars taken by Mariner 9 during Mars approach. The left frame is the actual image taken by Mariner 9 and the right frame is the predicted approach geometry. Original figures from [42], reprinted with permission of the American Institute of Aeronautics and Astronautics.

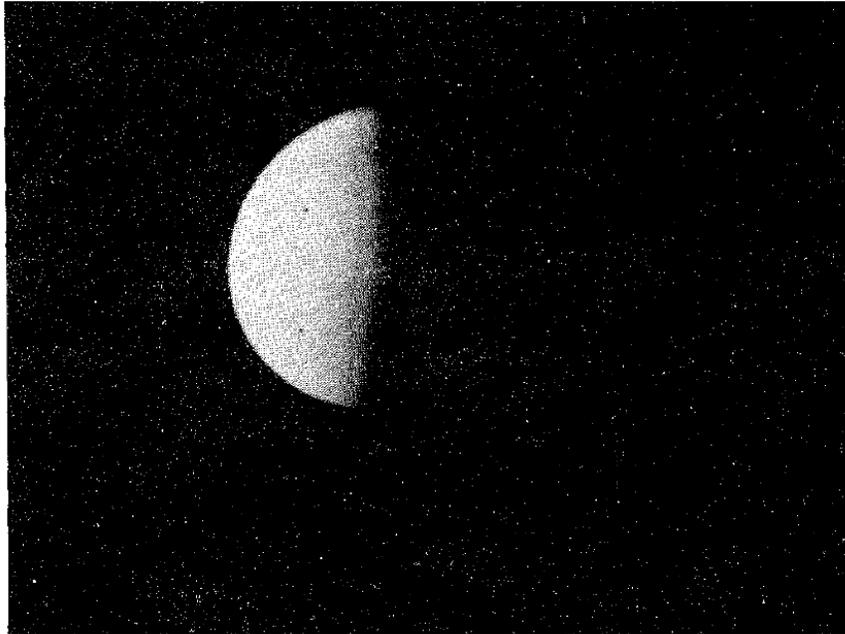


Figure 2.6: OPNAV image of Mars and background stars taken by Mariner 9. Original figure from [43], reprinted with permission of the American Institute of Aeronautics and Astronautics.

use of optical navigation in future robotic missions.

2.3.2 Voyager and Cassini

The Voyager missions (Voyager 1 launched on September 5, 1977, and Voyager 2 launched on August 20, 1977)[45, 46] and the Cassini/Huygens mission (launched on October 15, 1997)[47, 48] used similar optical navigation techniques and algorithms to meet mission navigational requirements. Although Mariner 9 provided a successful demonstration of optical navigation in 1971, the Voyager 1 and Voyager 2 encounters with Jupiter in 1979 were the first time that optical navigation was required for an interplanetary mission

to meet the principal mission objectives.[44, 49] These optical measurements were found to be important in the vicinity of the outer planets (and especially the moons of the outer planets) due to difficulties with precise flybys when planet/moon states are not well known. For Voyager, there were also significant *a-priori* uncertainties in the mass properties of the outer planets, especially Uranus and Neptune, leading to difficulties in accurately predicting the post-flyby trajectory of the spacecraft. Optical observations of a planet's satellites allowed for improved estimates of the planet's mass, thereby improving the quality of the spacecraft trajectory propagation. The apparent movement of the planet and moon(s) relative to the background starfield (and relative to each other) provide the information required to answer these questions. As an example, consider the changing geometry observed during Voyager's 1989 Neptune encounter, as shown in Fig. 2.7. The movement of Triton and Nereid (two moons of Neptune) relative to the planet was used to refine the orbits of these satellites and improve the estimate of the mass of Neptune. Simultaneously, measurements between these bodies and reference stars were used to provide information about the position of the spacecraft relative to the bodies in the Neptune system.

The OPNAV process for Voyager and Cassini may be broken into three components: (1) image planning, (2) image processing, and (3) orbit determination. The OPNAV images captured by the spacecraft were sent back to Earth for image processing. These results were then fused with ground-based tracking data and a state update was sent back to the spacecraft. Because a

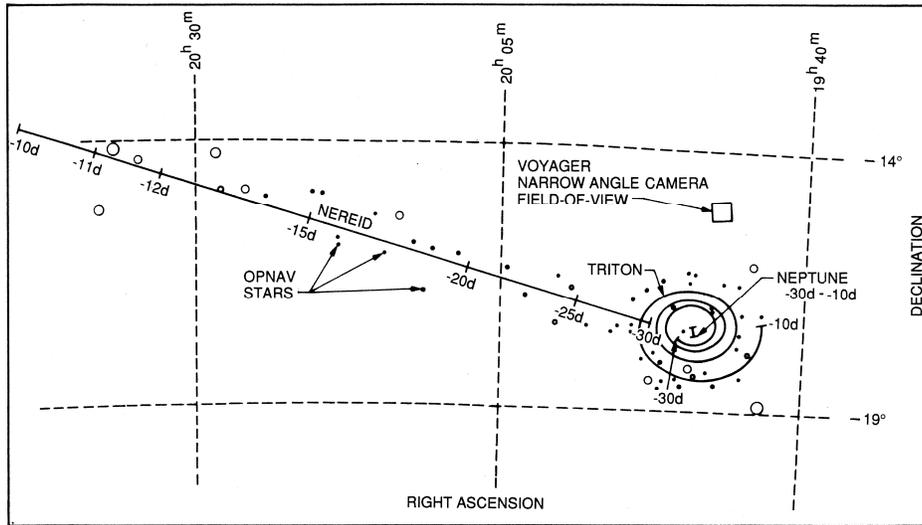


Figure 2.7: Neptune system and background star field viewed by Voyager on approach. Original figure from [50], reprinted with permission of the American Institute of Aeronautics and Astronautics.

large number of OPNAV images are required, this procedure can consume significant communication bandwidth. Voyager 2, for example, collected approximately 225 OPNAV images during its Uranus flyby[51] and approximately 450 OPNAV images during its Neptune flyby.[50] Cassini collected nearly 2250 OPNAV images between October 13, 2003 and March 23, 2007 (see Fig. 2.8).[52]. As an interesting aside, the Galileo spacecraft was originally going to use a similar approach, but the failure of the high gain antenna prevented the spacecraft from being able to transmit a sufficient number of OPNAV images back to Earth. The approach used on Galileo is discussed in a later section.

Image planning for Voyager and Cassini was performed on the ground well in advance of the flyby or maneuver of interest. Voyager 2 began image

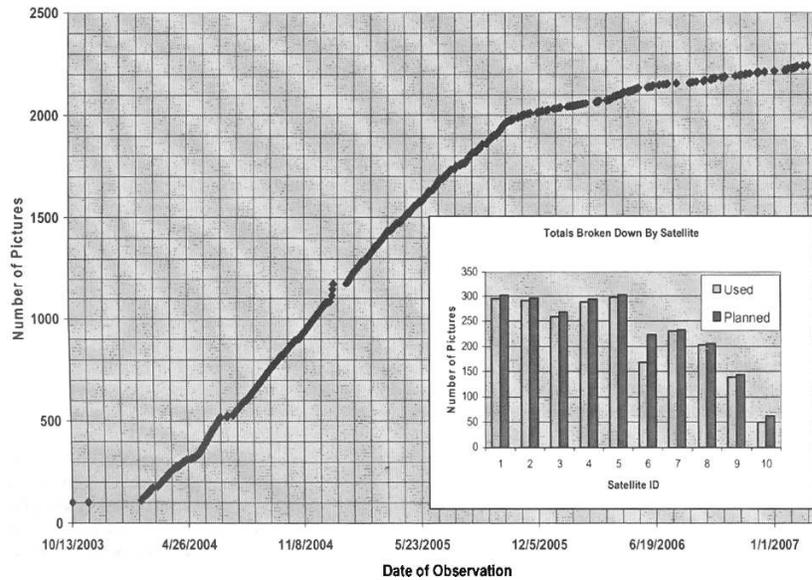


Figure 2.8: Cumulative distribution of Cassini OPNAV images. Original image from [52], reprinted with permission of the American Astronautical Society.

planning for the 1989 Neptune encounter over two years before the flyby,[50] and planning for an OPNAV imaging sequence could take up to six months for Cassini.[52] In both of these missions, OPNAV engineers were required to submit requests for spacecraft activity time. This was a rather complicated process. On Cassini, for example, the OPNAV engineers were one of 14 different groups competing for observing time on the spacecraft. These teams also had to determine how to downlink the OPNAV images through DSN. The Cassini mission found these competing interests could be best managed through a waypoint strategy for the sequence development process.[52]

After the OPNAV images were transmitted back to Earth via DSN, the image processing phase began. The first step is image registration, which

was performed manually. The analyst used a “drag-and-drop” approach to manually align the predicted location of objects with the actual observed location. This task typically resulted in a registration error on the order of a few pixels. For the Voyager OPNAV images, this process took about five minutes per image.[51]

After the manual image registration, filters were applied to determine the location of stars or planet/moon horizon with a subpixel accuracy. The centroid of a candidate star was found by using a Gaussian linear filter as a template - the Voyager OPNAV team called this “box filtering.” By taking the convolution of a normalized Gaussian kernel with a normalized patch from the image (normalizing the patch from the image makes this template matching problem insensitive to the brightness of the star), the centroid of a star may be accurately determined. This is a standard image processing technique that will be discussed in detail in Section 4.2.1.

A limb-scanning procedure was employed to accurately determine the location of the horizon or terminator of a planet or moon. The procedure begins by performing a series of scans radiating from the expected center of the planet/moon like the spokes of a wheel, as shown in Fig. 2.9.[51] Along any given scan direction, the scan residuals are computed. If partial derivatives of the limb location with respect to the observed planet centroid location are known, then the predicted brightness profile may be recursively updated to minimize the residuals. When this process is complete, accurate estimates of the planet/moon centroid and horizon/terminator are known.

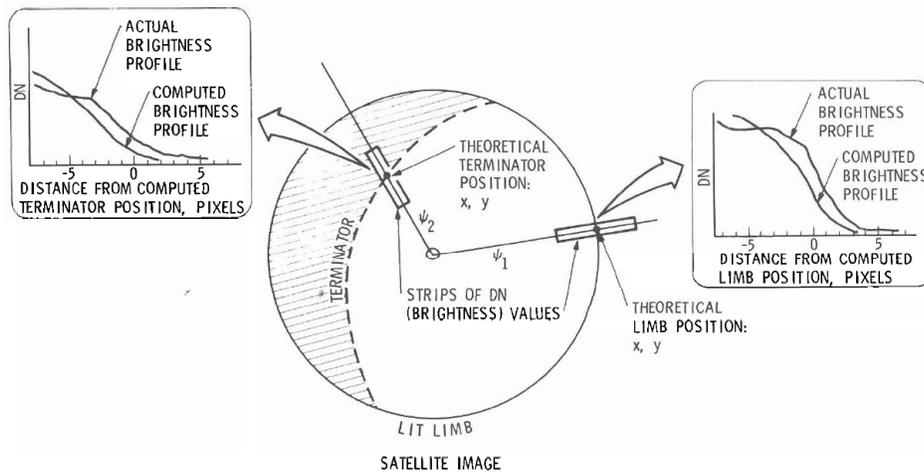


Figure 2.9: Voyager limb-scanning in the image plane. Original figure from [51], reprinted with permission of the American Institute of Aeronautics and Astronautics.

When star/planet/moon centroids and planet/moon horizons are known, all the information necessary to generate optical measurements is available. Specifically, Voyager and Cassini both used the angle between the planet/moon horizon and a reference star for optical navigation. Once these angular measurements were obtained, they were incorporated into a navigation filter. The mathematical relations required to do this are discussed in Chapter 4.

2.3.3 Galileo

The Galileo mission to the Jovian system launched onboard the Space Shuttle Atlantis (STS-34) on October 18, 1989. It arrived at Jupiter on December 7, 1995. The mission consisted of an orbiter and a probe that entered Jupiter's atmosphere. In April 1991 (during transit to Jupiter) the attempt to

deploy the orbiter high-gain antenna was unsuccessful.[53] Due to this failure, Galileo was forced to use its low-gain antenna for communications throughout the entire mission. The result was a significant reduction in communication bandwidth - the high-gain antenna was designed to transmit at data rates up to 134.4 kbps, while the low-gain antenna achieved a maximum data rate of only about 160 bps.[54] The optical navigation methodologies developed to deal with this reduced data rate provided important advances in image processing and in autonomy.

On the Galileo low-gain antenna, sending a single 800×800 pixel OPNAV image would take over 70 hours.[55] Therefore, it was clearly not feasible to take hundreds of OPNAV images as was done in Voyager and Cassini. This constraint led to the development of a new OPNAV technique called the single-frame mosaic (SFM). The SFM technique allowed for optical navigation with significantly fewer images. A single SFM image is created by leaving the camera shutter open as the spacecraft is slewed to a number of different attitudes. Consider, for example, the Galileo encounter with the asteroid 951 Gaspra (this was the first time a spacecraft had performed an encounter with an asteroid).[55] For this encounter, only four OPNAV images were performed. Reproductions of these four OPNAV images may be seen in Fig. 2.10.

Although some basic image smearing models were used for Voyager and Cassini, the image processing techniques for these missions were predicated on the assumption that stars/planets/moons remain stationary over a fixed set of pixels during the entire camera exposure.

The uncompensated wobble of the Galileo spacecraft and the slewing of the camera resulted in extended image trails (or patterns) for Gaspra and the background stars in the four SFM OPNAV images in Fig. 2.10. Within a

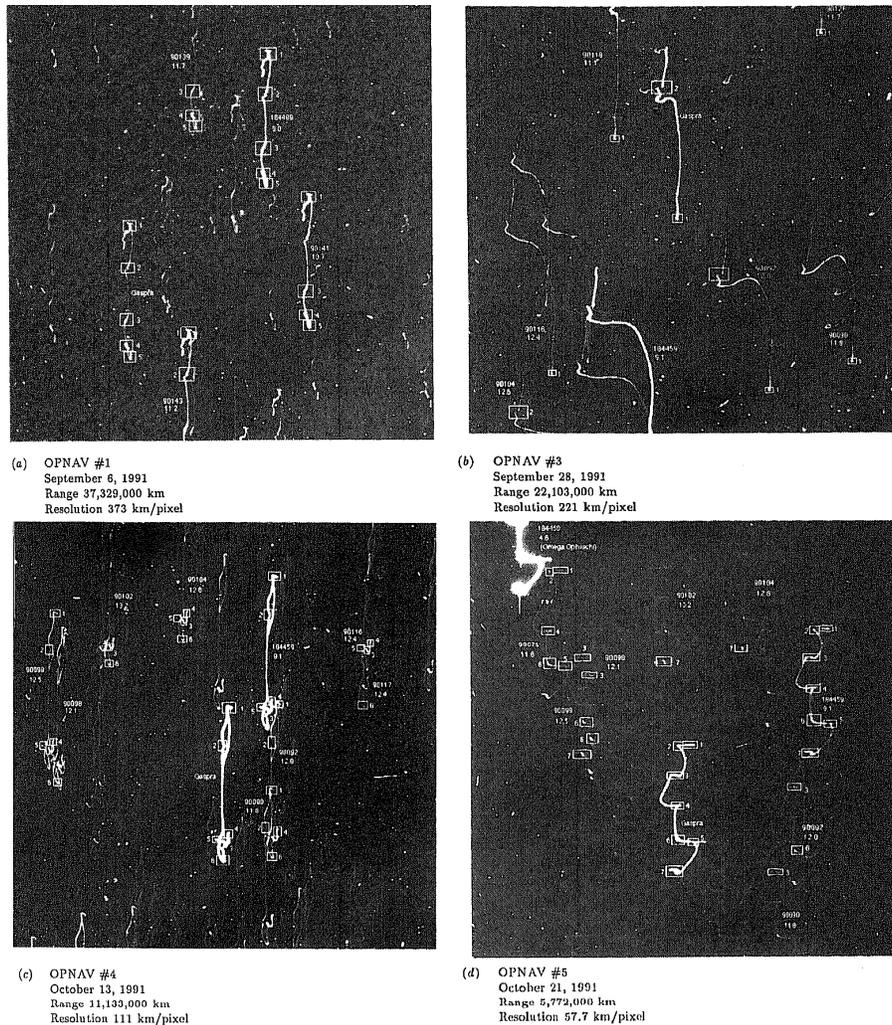


Figure 2.10: The four single-frame mosaic OPNAV pictures for the Galileo encounter with the asteroid 951 Gaspra. Original figure from [55], reprinted with permission of the American Institute of Aeronautics and Astronautics.

single image, all of these trails should have the same overall structure because they were created from the same camera motion. On Galileo, there was no simulation available that could accurately predict the pattern of the image trails in the SFM. Therefore, the actual patterns that were observed in the image, regardless of the conditions that led to the creation these patterns, were used to construct a template.[55]

A small characteristic part of the trails, called a “node,” would be selected as the template (or filter). This template was then compared to all the corresponding nodes on the other trails. This comparison was performed by taking the convolution of the normalized template with a normalized patch from the image and finding the template location that maximized the response. This process was repeated for each node by cycling the trail from which the template was created. Therefore, if n trails were identified, then there would be $n - 1$ estimates of the filter center location for each trail. Combining all the resulting filter centers for a single node resulted in an ensemble center location for that node (see Fig. 2.11).

Simple methods for determining the best ensemble center location, such as weighted averaging, were attempted but found to be insufficient. Instead, the analyst began the ensemble center finding process by visually selected a “nodal anchor point” at approximately the same location in each trail using a “point-and-click” approach. This nodal anchor point set up a local coordinate frame and could be related to the image origin (upper left corner) by a simple translation (see section 3.1.3 for more information on image coordinate

frames). A least squares approach was then used to simultaneously minimize the offset for all filter-image combinations. This procedure would come to be known as the multiple cross-correlation (MCC) algorithm. The result of the MCC algorithm is an estimate the ensemble center location for each trail with subpixel accuracy.

If each trail in an OPNAV image corresponds to a known celestial body or background star, then determining the ensemble center locations for a particular node describes the geometry of those bodies at a given epoch. This is the same information that would be extracted from a single “point-and-shoot” image (like what was used in Voyager and Cassini). By picking a different node, a different geometry will be observed. In this way, a SFM

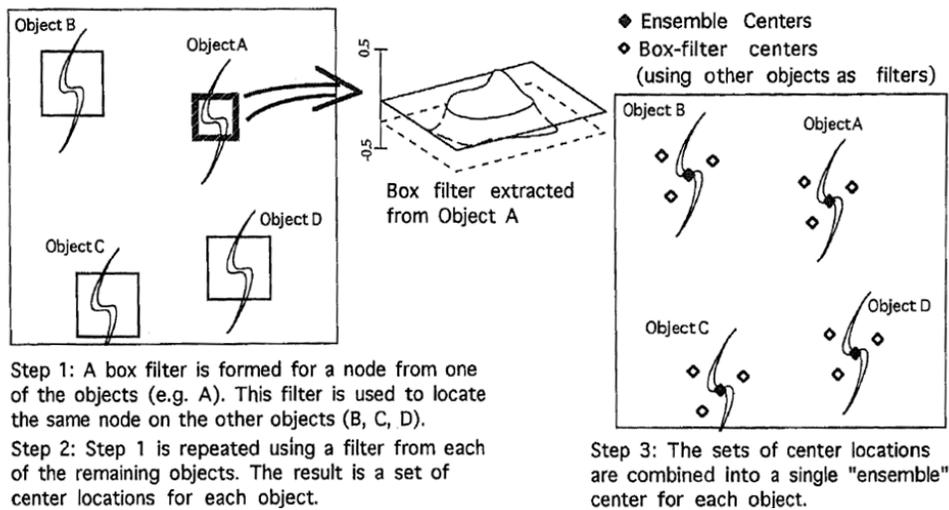


Figure 2.11: Centerfinding process used in the Galileo mission for a single node in a single-frame mosaic image. Original figure from [55], reprinted with permission of the American Institute of Aeronautics and Astronautics.

contains the same amount of information as many traditional images. With the geometry of the celestial body (e.g. asteroid) and background stars known at multiple epochs, angular measurements may be made and incorporated into a navigation filter.

It should be clear, however, that the SFM technique is only good when the target planet/moon/asteroid is unresolved in the image (i.e. its image subtends a few pixels at most). If an OPNAV image contains a celestial body that appears many pixels in diameter and is subjected to the slewing required to create an SFM image, then an accurate centroid or horizon location may not be found for the celestial body. Therefore, a new method was developed for Galileo's optical navigation within the Jovian system, where Jupiter or its moons appeared as large extended objects in the OPNAV images.

As with the interplanetary optical navigation, the low data rate prevented Galileo from sending a large number of OPNAV images back to Earth. This led Galileo engineers to automate part of the manual registration process described above for Galileo and Cassini (although Cassini was launched after Galileo, it still used manual registration). Then, only the portions of the OPNAV image that contained useful navigation information were transmitted to Earth. This reduced the amount of data that needed to be transmitted by a factor of 100 to 400.[56] The difficulty with automated registration lied with imaging the moons of Jupiter; the location of the stars in the OPNAV image could be predicted to within a few pixels from the attitude estimate provided by the star tracker. This need led to the development of the Galileo auto-rover

(AUTONomous Registering of a predict OVERlay) algorithm.[56]

The function of the auto-rover algorithm was primarily to search the image for extended objects. It then attempted to match the observed object with a predicted pattern of a moon limb. The algorithm began by scanning rows of the image, skipping a predetermined number of rows between each scan to increase algorithm speed. When a pixel with an intensity above a given threshold was detected, the surrounding pixels were surveyed to see if their intensity was above a minimum level and sustained for at least a minimum width. If these criteria were met, the pixel that first tripped the threshold was identified as a candidate limb location. This technique is a slightly more sophisticated version of the Mars limb finding algorithm developed for Mariner 9. Next, the auto-rover algorithm computed the predicted limb locations using the same brightness and width restrictions. The final step was an exercise in pattern recognition, with a goal of finding the best match between the predicted limb pattern and some ensemble of candidate limb locations extracted from the OPNAV image. What is clear, even without considering the mathematical details, is that the Galileo auto-rover algorithm required the *a-priori* position estimates of the spacecraft and of the moon to both be good enough to generate a predicted limb pattern that was close enough to the observed limb pattern to be successfully matched.

As an example, consider the auto-rover generated OPNAV image of Ganymede and a reference star taken by Galileo on June 3, 1996 shown in Fig. 2.12. Note that image content only exists along the limb and terminator

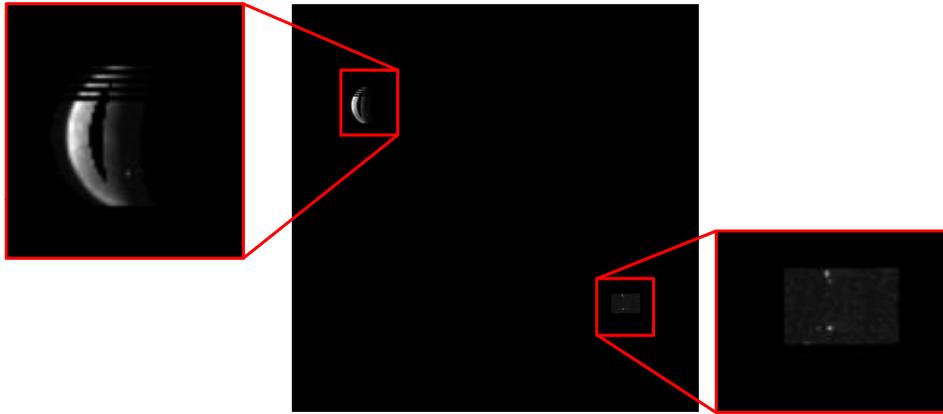


Figure 2.12: Galileo OPNAV image of Ganymede and reference star taken on 3 June 1996. Zoom-in of Ganymede on left and zoom-in of reference star on right. Original image from [57], courtesy of NASA/JPL-Caltech.

of Ganymede and around the reference star. This means that the auto-rover algorithm successfully identified Ganymede and the reference star in an autonomous fashion and only transmitted the relevant navigation data back to Earth. Sending the entire auto-rover generated OPNAV image shown in Fig. 2.12 required only 24,000 bits of data to be transmitted, while a full OPNAV image is about 5.12 million bits (800×800 pixel image with 8 bits per pixel).

2.3.4 Near Earth Asteroid Rendezvous (NEAR) Mission

The Near Earth Asteroid Rendezvous (NEAR) spacecraft was launched in February 1996 and inserted into orbit about the asteroid Eros in February 2000.[58] This made NEAR the first spacecraft to orbit an asteroid. Then, on 12 February 2001, the NEAR spacecraft landed on the surface of Eros.

The NEAR spacecraft used DSN tracking, a laser rangefinder, and

OPNAV images for navigation.[59] The OPNAV images of Eros captured by NEAR were sent back to Earth for ground-based processing. The irregular shape and unknown rotation of Eros made the star-horizon measurements used on previous missions impractical. During the approach phase, when the asteroid subtended only a few pixels, an analyst would find the asteroid center by eye and then manually move an overlay of the Eros limb so that it matched what was observed in the OPNAV image. As Eros became larger, and craters became visible, the OPNAV approach for NEAR switched to the tracking of surface features. These craters first became visible when the asteroid was about 30-40 pixels in diameter.[60] Once craters became visible, the observations were used to create a landmark (crater) database. Then, when an OPNAV image containing craters was received on the ground, an analyst would trace a cursor around the rim of each crater in the image to record crater rim points. An ellipse was then fit to these points. This information was used to match the observed crater to one of the craters in the database. Finally, using many such images spread over a period of time, the spacecraft's orbit and the rotation of Eros can be estimated. The NEAR spacecraft downlinked a total of 33,968 images to the ground, 17,601 of which contained at least one useful landmark for navigation.[60]

2.3.5 Deep Space 1 (DS1)

Deep Space 1 (DS1) was launched on October 24, 1998 as part of the NASA New Millennium Program.[61] The purpose of the New Millennium

Program was to provide a mechanism for testing high-risk, high-reward spacecraft technologies. One of the new technologies tested on DS1 was a new Autonomous Navigation (AutoNav) system. This new AutoNav system was required to (1) use optical images of distant asteroids for interplanetary orbit determination, (2) use the spacecraft ion propulsion system to control/maintain the spacecraft's orbit, (3) use optical images of the science target for approach orbit determination, and (4) provide updates of the target position after the flyby. The successful implementation of this technology made DS1 the first planetary mission to autonomously navigate all mission phases.[62]

During the interplanetary cruise phase, the distant asteroids used for navigation were very dim. Imaging these objects, therefore, required a long exposure time which led to unavoidable image smearing. As was noted in the Galileo mission, the patterns observed in the image smearing could not be predicted. Therefore DS1 used the SFM technique and MCC algorithm developed for the Galileo mission for autonomous interplanetary cruise navigation.[62]

During an encounter, the proximity of the asteroid and spacecraft caused the asteroid to appear much brighter than in the cruise OPNAV images. Therefore, the camera exposure time was significantly reduced to prevent pixel saturation. The resulting exposure times were short enough that smearing became unimportant. Unfortunately, the shape of the DS1 targets (mostly asteroids) were irregular and not well known. This precluded the use the horizon-star measurements that were the principal form of optical navigation during planetary encounters for Voyager, Cassini, and Galileo. Additionally,

because these encounters occurred at very high speed, the orbit determination algorithms used for interplanetary cruise were found to be too slow. DS1 addressed these problems by using the Reduced State Encounter Navigation (RSEN) algorithm during asteroid encounters.

The primary function of the RSEN algorithm during an encounter was to maintain a visual lock on the target asteroid throughout the flyby. Because the shape of the target asteroids were uncertain, tracking of the asteroid was performed using a simple center-of-brightness (COB) routine. This is much simpler (and faster) than the horizon filtering techniques used on previous planetary missions. The image processing overhead was further be reduced by only looking for the COB in the 3σ box that surrounded the predicted center. Here, it is worth noting that only the asteroid COB was used in RSEN. No stars were used in the algorithm, although the star tracker still provided attitude estimates. This means that the OPNAV data collected during an encounter was only a line-of-sight measurement to the asteroid COB. Therefore, the DS1 AutoNav system relied on the relative motion between the spacecraft and the asteroid for the changes geometry necessary for three-dimensional position information. Fortunately, the geometry changed very fast during these types of encounters.

To improve the speed of the estimation algorithm, the flyby was modeled as a straight-line trajectory, or simply

$$\mathbf{r}_{sc}(t) = \mathbf{r}_{sc}(t_0) + \mathbf{v}_{sc}(t_0) [t - t_0] \quad (2.1)$$

where $\mathbf{r}_{sc}(t_0)$ and $\mathbf{v}_{sc}(t_0)$ are the spacecraft position and velocity at the beginning of the RSEN sequence (usually about 30 minutes prior to the encounter). In the RSEN estimation process, the velocity knowledge was assumed to be perfect (and constant) and only updates to the spacecraft position estimate were performed. To simplify the dynamics, therefore, the RSEN algorithm relied on the assumption that all DS1 asteroid encounters were very fast flybys of a very small (low mass) objects. After the encounter, the DS1 AutoNav returned to the regular interplanetary navigation methodologies. As an interesting historical aside, the RSEN algorithm used on DS1 was an adaptation of the comet nucleus tracking algorithm designed for the Stardust mission's encounter with the comet Wild 2.[63] Although this technique was designed for Stardust, it actually flew on DS1 first.

The major advancements in the DS1 AutoNav system, therefore, were in the areas of automation rather than in image processing. The DS1 navigation system architecture (schematics may be found in [62] and [64]) was based on a derivative of the Mars Pathfinder flight software system.[62] Although Mars Pathfinder was controlled by long command sequences generated on the ground and uplinked to the spacecraft, it was adapted to meet the needs of DS1 by leaving large gaps in the uplinked command sequence. During the periods between the uplinked commands, the AutoNav system was responsible for autonomously completing the required tasks.[62]

Unlike Voyager, Cassini, and Galileo, image planning was performed autonomously on DS1. The DS1 AutoNav system, therefore, was required to

determine the schedule of targets. On the ground, a list of potential asteroid targets was generated as a function of mission time. Further, lighting constraints for the DS1 spacecraft created two natural clusters of these targets: one cluster in the “forward” anti-sun half-hemisphere and another cluster in the “aft” anti-sun half-hemisphere. To conserve fuel, the AutoNav system would only transition between these two clusters once (at most) during each OPNAV sequence. These clusters were predetermined on the ground. The clusters were also pre-ordered such that attitude maneuvers within each half-hemisphere cluster were minimized. The resulting ordered list was stored in the spacecraft’s memory. Despite having a well defined and ordered target list that was generated by analysts on the ground, the DS1 AutoNav’s picture planner was required to autonomously generate a sequence of specific image requests. Given this image sequence, it was also required to generate the turn commands necessary to rotate the spacecraft from one target to the next target. Once the OPNAV image is captured, the image processing is as described above.

The DS1 AutoNav system was later modified for use on the Deep Impact mission.[65]

2.3.6 MESSENGER

The MErcury Surface, Space ENvironment, GEochemistry and Rang-ing (MESSENGER) mission, built by the Johns Hopkins University Applied Physics Laboratory (JHU/APL), was launched on August 3, 2004.[66] This

mission was still ongoing at the time this dissertation was written. The interplanetary trajectory consists of one Earth flyby, two Venus flybys, and three Mercury flybys, prior to a planned orbit insertion about Mars in 2011. The early Earth and Venus flybys were used to test the cameras and OPNAV ground processing systems. The ground support for MESSENGER optical navigation included image planning, image scheduling/sequencing, image downloading, and operational interfaces.[66] The Earth and Venus flybys were not used for optical navigation because the atmosphere on these planets led to difficulties in measuring the horizon location. The first Mercury flyby, therefore, was the first time that OPNAV measurements were integrated into the orbit determination process.

As a representative example of an operational MESSENGER flyby, consider the first Mercury flyby, which occurred on January 14, 2008. Due to the possibility of significant errors in the *a-priori* Mercury state estimate, OPNAV data was needed to improve the estimate of the relative position of MESSENGER with respect to Mercury. This information allowed the MESSENGER navigation team to determine the current Mercury flyby altitude and, therefore, help design the trajectory correction maneuver (TCM) that would be used target the appropriate flyby altitude. To achieve this objective, a total of nine OPNAV imaging sequences were designed.[66] The first six imaging sequences provided enough information to narrow the TCM design space to just three options. The seventh and eighth OPNAV imaging sequences provided the data to finalize the TCM decision. The ninth OPNAV

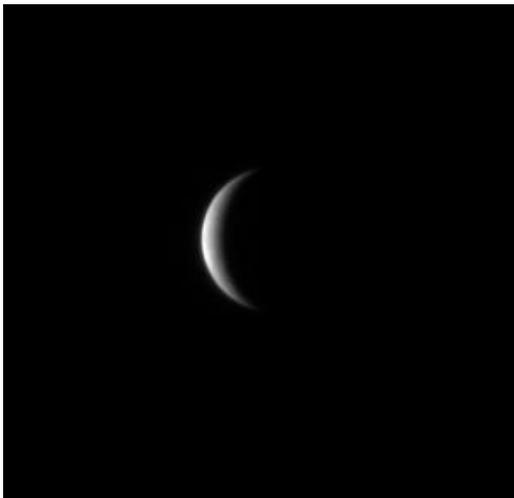


Figure 2.13: Example OPNAV image of Mercury taken by MESSENGER on 10 January 2008. Raw image is product ID CN0108443253M_IF_3 from [67]. Note that this image has been cropped to zoom in on Mercury.

imaging sequence occurred too late to support the TCM decision. Each of the nine OPNAV opportunities consisted of eight individual images taken in rapid succession.[66] Each of these eight images was taken by a different camera (narrow angle camera or wide angle camera), had a different exposure time, and had different pixel binning. An example image from the third OPNAV imaging sequence is shown in Fig. 2.13.

2.3.7 Robotic Missions to Mars

Most of the missions to Mars launched in the 1990s and 2000s relied solely on DSN based tracking for orbit determination and did not use optical navigation. Some of the notable missions that did not use any optical navigation include Mars Pathfinder,[1] Mars Odyssey,[68] the Mars Exploration

Rovers (MER),[69] and the Mars Phoenix Lander.[70] The unsuccessful Mars Climate Orbiter and Mars Polar Lander missions also did not use optical navigation of any kind.[1]

One of the instruments onboard the Mars Reconnaissance Orbiter (MRO), launched on August 12, 2005, was the Optical Navigation Camera. Although an optical navigation experiment was flown on MRO, the mission relied on DSN tracking for navigation and OPNAV images were not required to meet mission navigation requirements. The importance of this experiment lies primarily in the camera system, rather than the image processing techniques or navigation algorithms. This camera represents one of the first times that a dedicated OPNAV camera has been flown on a spacecraft. Previous missions have primarily relied on images captured by cameras that were designed to perform scientific experiments. Recent advancements have allowed for significant reductions in camera power and mass that made this experiment possible - the MRO optical navigation camera had a mass of 2.8 kg and a power requirement of 3-5 W.[3] An example image captured by the MRO's optical navigation camera is shown in Fig. 2.14. Note the significant improvements in the quality of this OPNAV image of Deimos relative to what was obtained using 1960s-1970s technology on Mariner 9 (see Fig. 2.5).

2.3.8 Robotic Missions to the Moon

Little use has been made of optical navigation by robotic spacecraft in cislunar space and in low Lunar orbit (LLO). The Ranger,[72] Surveyor,[73]

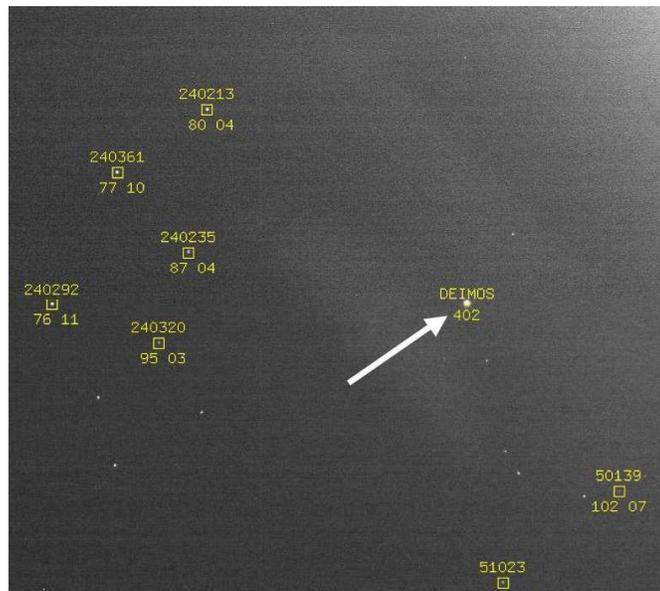


Figure 2.14: Mars Reconnaissance Orbiter OPNAV image of Deimos and reference stars taken on 6 March 2006. Original image from [71], courtesy of NASA/JPL-Caltech.

and Lunar Prospector[74] missions all only used radiometric tracking.

The Clementine spacecraft, launched on January 25, 1994, also only used radiometric tracking. There were plans for the Clementine mission to perform autonomous optical navigation, but a software failure occurred in May 1994 and spacecraft operations were terminated in August 1994 - before the autonomous navigation algorithms were ever used in orbit.[75] These algorithms, however, were tested extensively on the ground using actual images obtained from Clementine during the LLO phase of the mission.[76]

The Lunar Reconnaissance Orbiter (LRO) was launched on June 18, 2009, with the objective of providing high-resolution topographic maps and

images of the lunar surface.[77] LRO uses radiometric tracking in conjunction with measurements from the Lunar Orbiter Laser Altimeter (LOLA) science instrument for orbit determination. There are no plans to use optical navigation on the LRO mission.

Chapter 3

Image Creation and Optical Sensors

3.1 Rotations, Transformations, and Reference Frames

3.1.1 Basic Attitude Representations

What follows is a brief review of key attitude representation relationships and a description of the attitude nomenclature to be used in the subsequent discussions. Numerous references discuss this subject in substantially greater detail.[78–80]

Begin by recalling that any rotation may be defined as a single axis rotation. Let \mathbf{e}_θ be the axis of rotation (or “Euler Axis”) and let θ be the angle of rotation about this axis (or “Euler Angle”). This rotation may also be expressed in matrix form, \mathbf{T} , as a Direction Cosine Matrix (DCM, sometimes called a rotation matrix):[81]

$$\mathbf{T} = (\cos\theta)\mathbf{I}_{3\times 3} + (1 - \cos\theta)\mathbf{e}_\theta\mathbf{e}_\theta^T - (\sin\theta) [\mathbf{e}_\theta \times] \quad (3.1)$$

$$\mathbf{T}\mathbf{T}^{-1} = \mathbf{I}_{3\times 3} \quad \text{and} \quad \mathbf{T}^{-1} = \mathbf{T}^T \quad (3.2)$$

where the nomenclature $[\boldsymbol{\alpha} \times]$ is defined as the skew-symmetric cross product matrix such that $\boldsymbol{\alpha} \times \boldsymbol{\beta} = [\boldsymbol{\alpha} \times] \boldsymbol{\beta}$,

$$[\boldsymbol{\alpha} \times] = \begin{bmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{bmatrix} \quad (3.3)$$

Further, note that Eq. 3.1 may be rewritten in matrix exponential form as [82]

$$\mathbf{T} = \exp \{-\theta [\mathbf{e}_\theta \times]\} \quad (3.4)$$

and under a small angle rotation, $\delta\theta \mathbf{e}_{\delta\theta} = \delta\boldsymbol{\theta}$, where only first order terms are kept, this becomes

$$\mathbf{T} = \exp \{-\delta\theta [\mathbf{e}_{\delta\theta} \times]\} \approx \mathbf{I}_{3 \times 3} - \delta\theta [\mathbf{e}_{\delta\theta} \times] \quad (3.5)$$

In order to clearly indicate the direction of rotation, the following notation is introduced. The rotation matrix \mathbf{T}_B^A represents a rotation from the A frame to the B frame. In other words,

$$\mathbf{r}_B = \mathbf{T}_B^A \mathbf{r}_A \quad (3.6)$$

where \mathbf{r}_A is a vector expressed in the A frame and \mathbf{r}_B is the same vector expressed in the B frame.

A useful three-dimensional representation is given by Rodrigues parameters (frequently also referred to as the Gibbs vector). The idea of a three dimensional parameterization of the attitude follows directly from Euler axis/angle discussion above. Let the Gibbs vector, \mathbf{g} , be defined as

$$\mathbf{g} = \left(\tan \frac{\theta}{2} \right) \mathbf{e}_\theta \quad (3.7)$$

Further, two sequential rotations (first by \mathbf{g}_1 and then by \mathbf{g}_2) expressed as Gibbs vectors may be combined through the composition formula:[83]

$$\mathbf{g}_3 = \frac{\mathbf{g}_1 + \mathbf{g}_2 - \mathbf{g}_2 \times \mathbf{g}_1}{1 - \mathbf{g}_1^T \mathbf{g}_2} \quad (3.8)$$

An alternate way to express attitude is through the quaternion. The quaternion, $\bar{\mathbf{q}}$, is a four-element parameterization of the attitude and is given by:

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} \sin \frac{\theta}{2} \mathbf{e}_\theta \\ \cos \frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (3.9)$$

$$\bar{\mathbf{q}}^T \bar{\mathbf{q}} = 1 \quad (3.10)$$

This quaternion may be related to the equivalent DCM attitude representation through[78]

$$\mathbf{T}(\bar{\mathbf{q}}) = (q_4^2 - \mathbf{q}^T \mathbf{q}) \mathbf{I}_{3 \times 3} + 2\mathbf{q}\mathbf{q}^T - 2q_4 [\mathbf{q} \times] \quad (3.11)$$

Quaternion mathematics, developed by Sir William Rowan Hamilton in the 1800s, provides the tools necessary to perform sequential rotations directly on the quaternion. A detailed review of some of the key results from quaternion mathematics is provided in Appendix A. The quaternion product,¹ denoted by the symbol \otimes , is given by

$$\bar{\mathbf{q}} \otimes \bar{\mathbf{p}} = \begin{bmatrix} q_4 \mathbf{p} + p_4 \mathbf{q} - \mathbf{q} \times \mathbf{p} \\ q_4 p_4 - \mathbf{q}^T \mathbf{p} \end{bmatrix} \quad (3.12)$$

The quaternion product as shown in Eq. 3.12 allows for sequential rotations in the same manner as DCMs:

$$\mathbf{T}_C^A = \mathbf{T}_C^B \mathbf{T}_B^A = \mathbf{T}(\bar{\mathbf{q}}_C^B) \mathbf{T}(\bar{\mathbf{q}}_B^A) = \mathbf{T}(\bar{\mathbf{q}}_C^B \otimes \bar{\mathbf{q}}_B^A) = \mathbf{T}(\bar{\mathbf{q}}_C^A) \quad (3.13)$$

¹Note that this quaternion product is backwards when compared to the classical quaternion product, as is standard practice in the attitude determination community. This is done so that the order of quaternion multiplication mirrors the order of DCM multiplication for sequential rotations. If the classical quaternion product is denoted by \odot , then $\bar{\mathbf{q}} \otimes \bar{\mathbf{p}} = \bar{\mathbf{p}} \odot \bar{\mathbf{q}}$

3.1.2 Homogeneous Coordinates

Homogeneous coordinates allow for rotation and translation to be simultaneously performed in a single operation and with compact notation. They also allow for transformations in projective space, just as Cartesian coordinates allow for transformations in Euclidean space.[84, 85]

Homogeneous coordinates are created by increasing the dimensionality of the Cartesian coordinates by one. This additional coordinate is a scale factor. Therefore, to create homogeneous coordinates from a set of given Cartesian coordinates, the scale factor is set to one (because the given Cartesian coordinates are already to scale):

$$\begin{bmatrix} X_h \\ Y_h \\ Z_h \\ W_h \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.14)$$

Now, assume that an object's position in frame A is given by \mathbf{p}_A . Further assume there is a second frame, B , which has an orientation relative to A given by \mathbf{T}_B^A . The origin of B is also displaced relative to the origin of A . Therefore, let the position of B relative to A , as expressed in frame A , be given by \mathbf{b}_A . Using homogeneous coordinates, the transformation of any position coordinate from A to B may be expressed in a single linear transformation,

$$\begin{bmatrix} \mathbf{p}_B \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{T}_B^A & -\mathbf{T}_B^A \mathbf{b}_A \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_A \\ 1 \end{bmatrix} \quad (3.15)$$

where \mathbf{p}_B is the position coordinate of the same object in frame B . Because no scaling is performed for this simple rotation and translation operation, the

scale factor remains 1 after the linear operation. This is not generally true for transformations with homogeneous coordinates, as is the case when they are used for perspective projection calculations.[84, 85] When the scale factor is not 1, the homogeneous coordinates must be scaled by $1/W_h$ to recover the corresponding Cartesian coordinates.

3.1.3 Discussion of Coordinate Frames

Begin by defining an inertial frame, I . For the purposes of the following discussions, unless otherwise stated, frame I is assumed to have its origin at the barycenter of the system in which the spacecraft orbit is described (e.g. solar system barycenter for heliocentric orbits, Earth-Moon barycenter for Earth/cislunar orbits, etc.) and is assumed to be a right-handed coordinate system. Also, define another inertial frame, I' , whose origin coincides with the spacecraft center of mass at a given epoch and whose axes are parallel to those of frame I .

Now define a non-inertial, right-handed coordinate frame attached to the spacecraft called the body frame and denoted by B . This frame has an origin at the spacecraft center of mass and axes related to the inertial frame by the rotation matrix \mathbf{T}_I^B . If a camera is mounted to the spacecraft, let the camera sensor frame be denoted by C . Further, if the camera is rigidly mounted to the spacecraft, the rotation matrix \mathbf{T}_B^C is assumed to be constant. The axes of C are chosen such that the Z -axis lies along the camera optical axis and the X -axis is the projection of the true detector x -axis in the plane

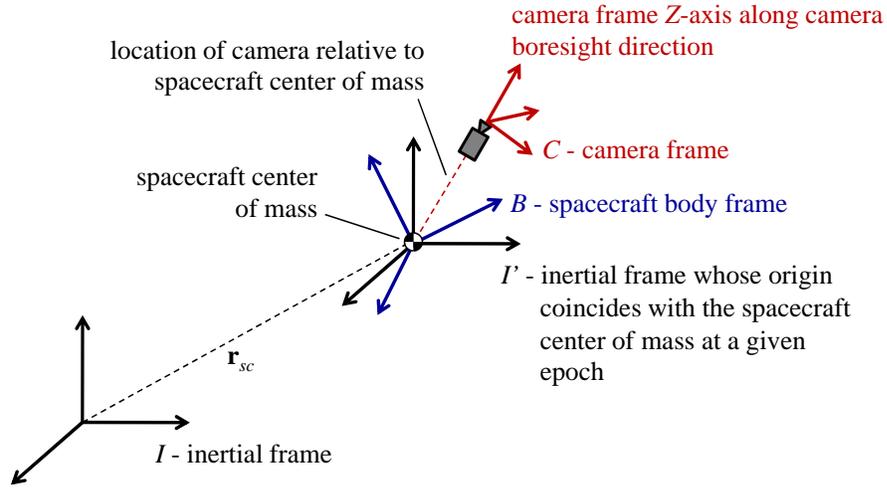


Figure 3.1: Graphical depiction of the relationship between the I , I' , B , and C frames.

perpendicular to the optical axis. The Y -axis completes the right-hand system. A point in frame C is denoted by $\mathbf{p}_C = [X \ Y \ Z]^T$. The relationship between the I , I' , B , and C frames is shown in Fig. 3.1.

Errors in construction of the camera require the introduction of an additional frame. Let D denote the detector frame. The origin of this frame is at the true detector principal point and the x - y plane lies in the true plane of the detector, which is likely to be offset and/or rotated with respect to the X - Y plane in frame C . The z -axis of frame D completes the right-handed system and is perpendicular to the true detector plane. Therefore, the difference between the Z -axis in the C frame and the z -axis in the D frame may be used to describe the detector plane misalignment. A point in frame D is denoted by $\mathbf{p}_D = [x \ y \ z]^T$.

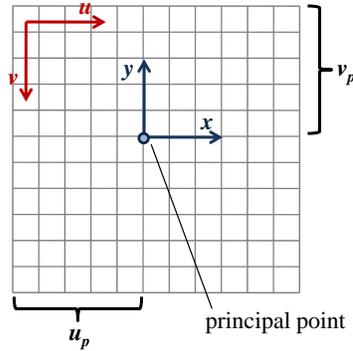


Figure 3.2: Detector coordinate frames.

The five frames introduced above (I , I' , B , C , and D) all exist in \mathbb{R}^3 . The image captured by the camera, however, must exist only in \mathbb{R}^2 . Therefore, let the two-dimensional array frame be denoted by $A \in \mathbb{R}^2$. As is standard practice for CCD/CMOS arrays, let the origin of A be in the upper left-hand corner of the array and let any point on the array be defined by the coordinates $[u, v]$. A graphical depiction of the relationship between frames D and A is shown in Fig. 3.2. Further, let the $[u, v]$ coordinates be expressed in units of pixels such that integer values of $[u, v]$ correspond to the center of the indicated pixel. If the array u -axis and v -axis are not perfectly perpendicular, a skew factor must be introduced. Here, define θ as the angle between the u -axis and v -axis (a perfectly constructed array will have $\theta = 90^\circ$). Therefore, any set of $[x, y]$ coordinates in frame D may be converted to a set of corresponding $[u, v]$ coordinates in frame A through the following simple transformation with homogeneous coordinates[86, 87]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & \cot(\theta) & u_p \\ 0 & \frac{-s_y}{\sin(\theta)} & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.16)$$

where s_x and s_y are the x -axis and y -axis scale factors in units of pixel/m, and $[u_p, v_p]$ is the CCD/CMOS principal point (i.e. the location of the intersection of the optical axis with the detector plane) expressed in the $[u, v]$ coordinate frame. Note that if $s_x = s_y$ then the pixels are square. If the mapping from D to A (given by the matrix in Eq. 3.16) is defined as \mathbf{M}_A^D , then the mapping from A to D is given by $\mathbf{M}_D^A = (\mathbf{M}_A^D)^{-1}$:

$$\mathbf{M}_D^A = \begin{bmatrix} \frac{1}{s_x} & \frac{\cos(\theta)}{s_x s_y} & -\frac{v_p \cos(\theta) + s_y u_p}{s_x s_y} \\ 0 & -\frac{\sin(\theta)}{s_y} & \frac{v_p \sin(\theta)}{s_y} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

Note that, despite the notation, \mathbf{M}_A^D and \mathbf{M}_D^A are not rotation matrices.

3.2 Formal Definition of an Image

An image is the projection of the three-dimensional world onto a two-dimensional space. In an optical system, information about the outside world is recorded as a two-dimensional surface of light intensity. Any infinitesimal patch of detector surface centered at $[x, y]$ may collect a number of photons over a given exposure duration. In an ideal system, the values of x and y may be treated as continuous and there are an infinite number of possible locations for this infinitesimal patch. Therefore, define the ideal image as a map, I_{ideal} , such that

$$I_{ideal} : \Omega_{xy} \subset \mathbb{R}^2 \rightarrow \mathbb{N}; \quad (x, y) \mapsto I_{ideal}(x, y) \quad (3.18)$$

where Ω_{xy} is the compact set that describes all possible points on the detector surface, i.e. $[x, y] \in \Omega_{xy}$. Additionally, let \mathbb{N} represent the set of all nonnegative integers, i.e. $\mathbb{N} = [0, 1, 2, \dots]$. Note that the $I_{ideal}(x, y)$ must exist in \mathbb{N} because only a discrete number photons can be detected.

Unfortunately, if the detector is a CCD or CMOS array, the domain in Eq. 3.18 must be discretized. To borrow the notation of Ma et al,[86] define a digital image as a two-dimensional brightness array, such that the image I_{dig} is given by

$$I_{dig} : \Omega_{uv} \subset \mathbb{N}^2 \rightarrow \mathbb{N}; \quad (u, v) \mapsto I_{dig}(u, v) \quad (3.19)$$

where Ω_{uv} is the compact set that describes all possible pixel locations on the detector surface, i.e. $[u, v] \in \Omega_{uv}$. It is interesting to note that Eq. 3.19 states that $\Omega_{uv} \subset \mathbb{N}^2$. Generally speaking, the variables u and v are continuous and some things (such as the principal point location) may be described at fractional values of u and v (in other words, one would usually say $[u, v] \in \Psi_{uv}$, $\Psi_{uv} \subset \mathbb{R}^2$). However, because digital image information only exists at pixel locations, only nonnegative integer values of u and v may be used to describe a digital image.

3.3 The Ideal Camera Model

The ideal camera model describes the geometry of image formation in a camera with no internal error sources. A camera is an optical device that, at a minimum, usually consists of a set of optics (lenses and/or mirrors) and a detector. Real cameras often require many other components, such as

shutters or light baffles, but these components need not be considered for the development of this simple camera model.

Begin the discussion of the ideal camera model by considering the geometry of a thin lens as shown in Fig. 3.3. Now, recall the thin lens equation,[88]

$$\frac{1}{s_1} + \frac{1}{s_2} = (n - 1) \left(\frac{1}{R_1} - \frac{1}{R_2} \right) = \frac{1}{f} \quad (3.20)$$

where s_1 is the distance between the actual object and the lens, s_2 is the distance between the lens and the focused image, f is the focal length, n is the index of refraction, R_1 is the radius of curvature of the lens on the side with the object, and R_2 is the radius of curvature of the lens on the side with the image.

Now, consider the notional view of a single thin lens viewing two objects infinitely far away shown in Fig. 3.4. One of these objects in Fig. 3.4 is along the optical axis and the other is not. As is clear from Eq. 3.20, the lens will create a focused image of an object that is infinitely far away

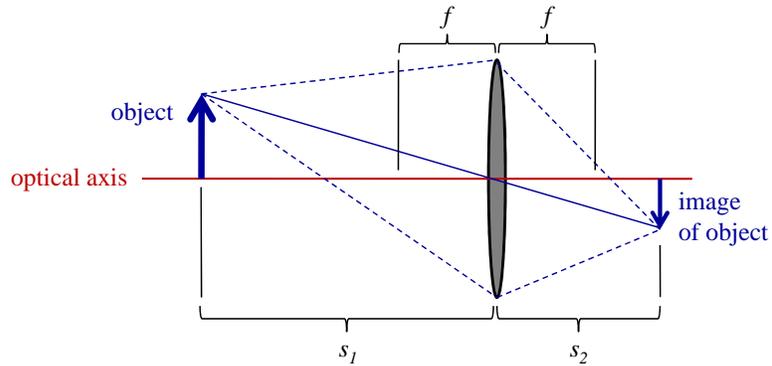


Figure 3.3: Diagram thin lens geometry.

when the separation between the lens and the detector is equal to the focal length. Additionally note that a single lens system will create a flipped image of the true geometry. In an idealized camera system, a ray of light that passes through the center of the aperture will be undeflected. Therefore, for the purposes of determining the geometric center of a focused point source, let the aperture of the camera become extremely small so that only the single undistorted ray that passes through the center of the lens is allowed to pass through the system. This idealized system is frequently called a “pinhole” camera model. Note, of course, that the pinhole model is good only for simple camera geometry calculations. A real system will not use a pinhole camera because the aperture must be large enough to allow a sufficient number photons into the system so that the detector can measure the observed scene with an acceptable signal-to-noise ratio.

Real cameras will also probably use a system of multiple lenses and mirrors, rather than just one simple thin lens. Single lens systems tend to

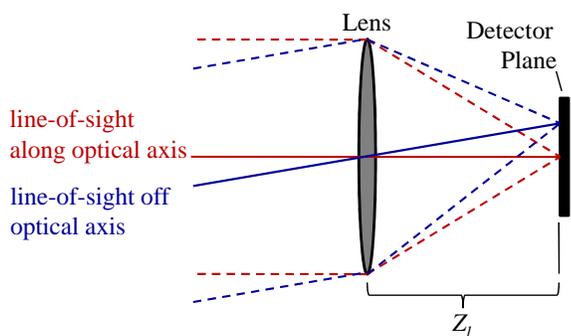


Figure 3.4: Diagram of light rays from two objects at infinity through a simple thin lens system.

exhibit significant aberrations and be unnecessarily large. A system of lenses and mirrors may be used to address both of these problems. With regard to aberrations, recall that although the image surface formed by the primary lenses/mirrors is curved, the detector is usually flat. Therefore, many telescopes and cameras, especially those with a wide field-of-view, place a set of auxiliary optics near the image plane to flatten the curved image plane onto the detector. These optical assemblies are typically called field flatteners or field correctors. More details on field flatteners may be found in [88]. To provide a concrete example, consider the optical layout for the Mars Reconnaissance Orbiter’s optical navigation camera shown in Fig. 3.5 (additional specifications provided in Table 3.1).

A system of mirrors and lenses also allows for a more compact optical device. Any system of mirrors and lenses may be shown to have an effective focal length (the system shown in Fig. 3.5 has an effective focal length of 50 cm). The geometry of image formation in such a system is the same as if the

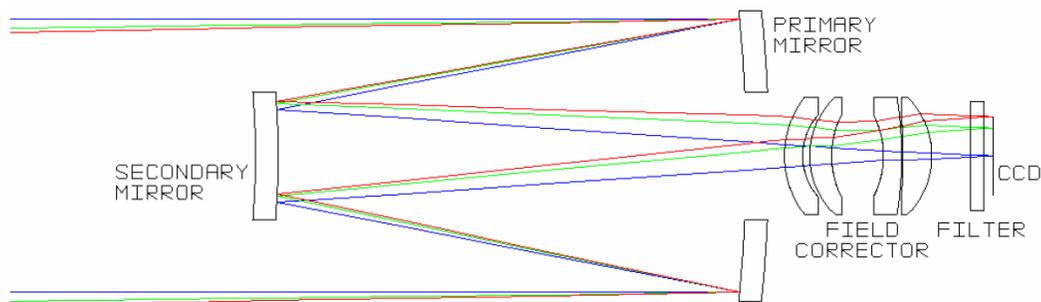


Figure 3.5: Optical layout for the Mars Reconnaissance Orbiter’s optical navigation camera. Original image from [89], reprinted with permission of SPIE and the authors.

Table 3.1: Summary of performance specifications for the Mars Reconnaissance Orbiter’s optical navigation camera.[89, 90]

Parameter	Description
Optics	Ritchey-Chrétin with refractive field corrector
Focal length	50 cm
Aperture	6 cm
F-number	F/8.33
Field-of-view	1.4 deg × 1.4 deg
Filter	Blocks wavelengths greater than 650 nm
Detector	CCD

system consisted of a single thin lens with a focal length equal to the equivalent focal length. A system of two thin lenses, for example, may be shown to have an effective focal length, f_{eq} , given by

$$\frac{1}{f_{eq}} = \frac{1}{f_1} + \frac{1}{f_2} - \frac{d}{f_1 f_2} \quad (3.21)$$

where f_1 is the focal length of the first lens, f_2 is the focal length of the second lens, and d is the distance between the two lenses. Therefore, the pinhole camera model may be used to accurately describe the geometry of image formation in a real optical system. Although a detailed discussion of optics and systems of multiple lenses is not provided here, a great deal of information on the topic is readily available in the literature.[88]

The pinhole camera model allows for a simple relationship between a point on the detector plane and the corresponding line-of-sight unit vector. A three-dimensional representation of this situation is shown pictorially in Fig. 3.6. Through simple trigonometry (similar triangles is the easiest method), the relationship between the lens-detector separation, a measured point on

the focal plane, and the line-of-sight unit vector to the image source is given by:

$$(\mathbf{e}_i)_C = \frac{1}{\sqrt{X_i^2 + Y_i^2 + Z_l^2}} \begin{bmatrix} -X_i \\ -Y_i \\ Z_l \end{bmatrix} \quad (3.22)$$

where X_i and Y_i are the coordinates where the central ray of the i -th observed point source pierces the XY plane of C . Note that the variable Z_l represents the effective separation between the lens and the XY plane of C . For a perfectly focused system (observing an object at infinity), Z_l is equal to the focal length. Additionally, note that Z_l is a constant for the system and is the same regardless of the object considered. For many applications, it may be necessary to rotate the line-of-sight vector from the camera frame to the body frame:

$$(\mathbf{e}_i)_B = \mathbf{T}_B^C (\mathbf{e}_i)_C \quad (3.23)$$

Again, note that while the simple pinhole camera model described above creates a flipped image, it may be advantageous to deal with an un-

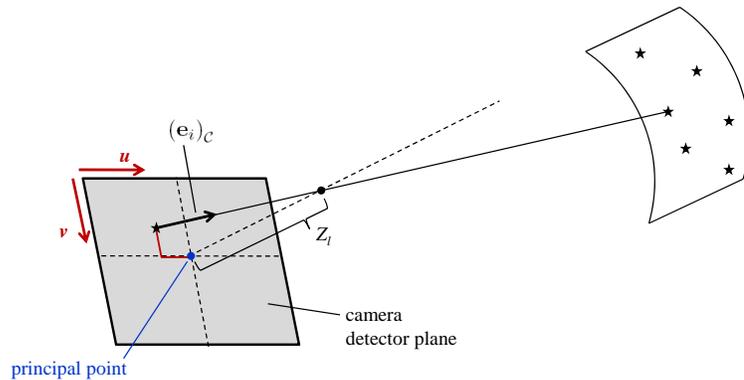


Figure 3.6: Line-of-sight geometry for a simple camera model.

flipped image. Such an image would be created if the three-dimensional world were projected onto an imaginary plane that lies in front of the lens and is parallel to the actual detector plane.[86, 91] This imaginary plane is frequently called the “frontal image plane” and has geometry as shown in Fig. 3.7. The separation between the frontal image plane and the lens is the same distance as (but in the opposite direction of) the separation between the detector plane and the lens. This will recreate an image that looks similar to what would be seen by a human with their eye located at the lens. For this reason, images are typically displayed in the frontal image plane, instead of in the configuration actually seen by the detector.

To maintain a consistent nomenclature, define the frontal image plane coordinates as $[u', v']$ with the origin in the upper left hand corner as seen from an observer located at the lens. This leads to the following relationship in homogeneous coordinates between an image on the actual detector plane and the corresponding image on the frontal image plane:

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & v_{max} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (3.24)$$

which is equivalent to

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & v_{max} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (3.25)$$

where v_{max} is the v -coordinate of the pixels in the bottom row of the CCD/CMOS detector. Therefore, a simple vertical flip is all that is required to transform

a digital image between $[u', v']$ coordinates in the the frontal image plane and $[u, v]$ coordinates on the actual detector.

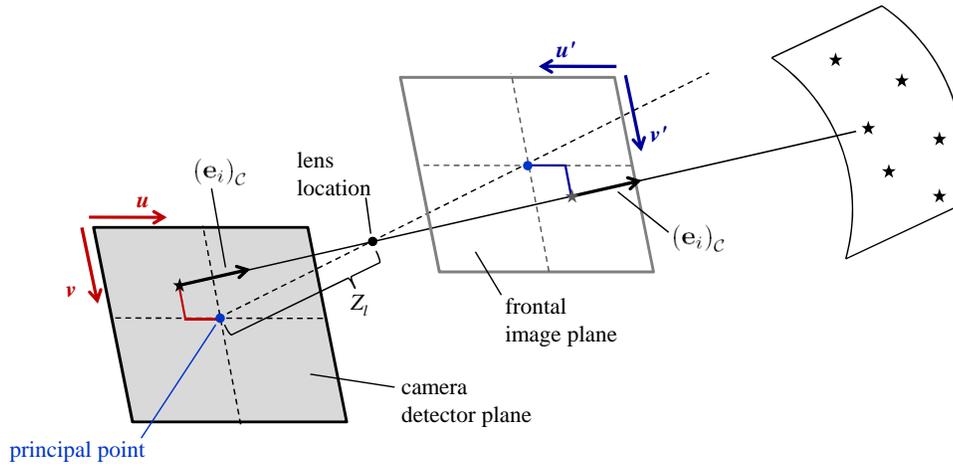


Figure 3.7: Camera model with frontal image plane.

3.4 Comments on Detector Hardware

3.4.1 Charged-Coupled Devices (CCD)

Most CCD sensors consist of an array of photoactive metal-oxide semiconductor (MOS) capacitors connected through a system of shift registers. These MOS capacitors are usually arranged in a grid pattern. When a photon with an energy greater than the energy gap is absorbed by one of these MOS capacitors, an electron-hole pair is created. When a voltage is applied, the electrons stay in the depletion region, while the holes move towards the ground electrode. Although the depletion regions may hold a very large number of electrons, their capacity is not unlimited. The pixel well capacity refers to the maximum number of electrons that may be stored in the depletion re-

gion. When the well capacity is reached, and no more electrons can be stored, the pixel is said to be saturated.

Therefore, to create an image, the shutter of the camera is opened and photons begin striking the CCD array. The charge in each well at each pixel location then increases linearly with time and the intensity of the light at that pixel location. After a period of time, the shutter is closed and the charges in each well must be output in a usable fashion.

The pixels in a CCD array are usually connected by a system of shift registers. If the depletion region of adjacent pixels overlap (either by row or by column), then charge may be shifted between pixels by controlling the voltage applied to the MOS capacitors. These capacitors, therefore, act as gates and are used to transfer charges across rows (or columns) of the array. The sequence of events described in Fig. 3.8 notionally shows how a charge may be transferred from one pixel to its neighboring pixel. Each row (or column) is shifted in parallel by one gate, and the last pixel in each row (or column) outputs in parallel to a serial shift register. This serial shift register is then read out. Once the entire serial readout register has been emptied, each row (or column) is shifted in parallel by one more gate. This process is repeated until the entire image is read out. The serial stream of charge packets in the readout register is sent to an output circuit where the charge packets are converted to a voltage.

Because the depletion region of adjacent pixels overlap, excess charge generated from a saturated pixel may bleed over into neighboring pixels in

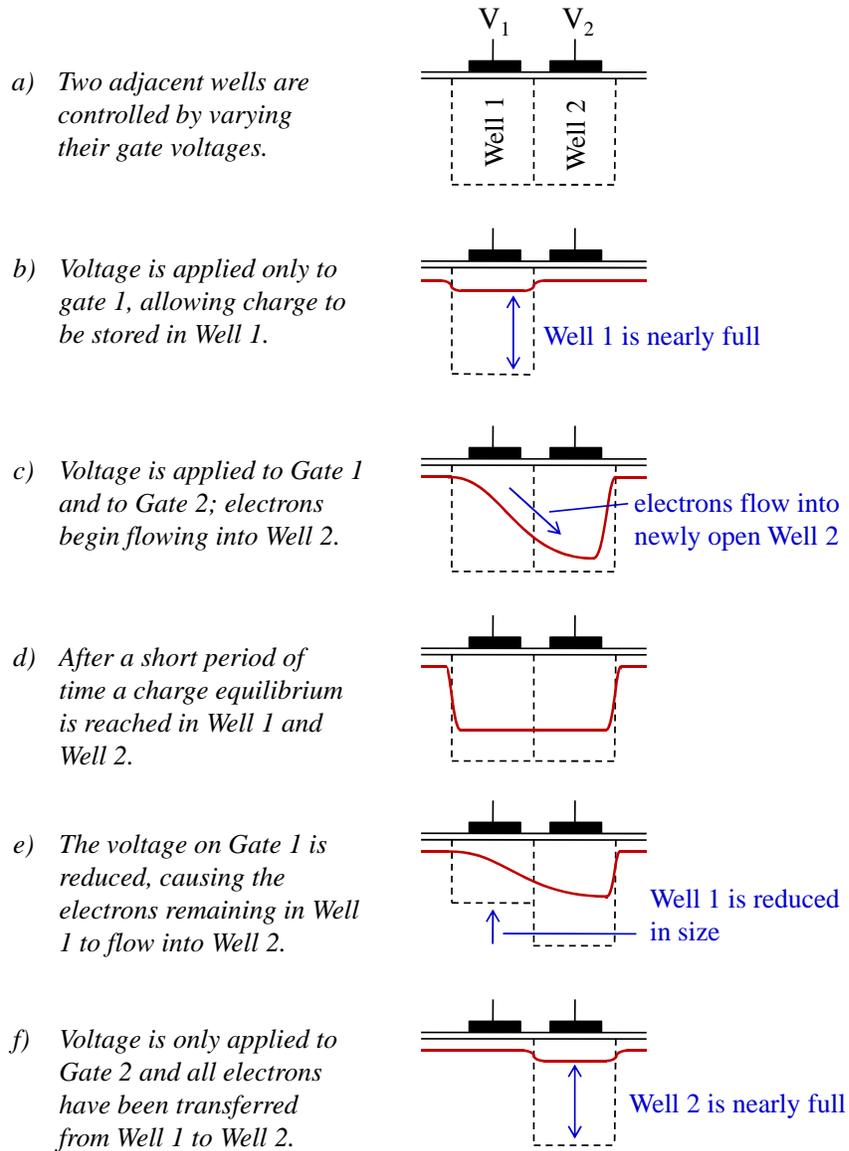


Figure 3.8: Notional diagram for the process of transferring electrons between adjacent wells in a CCD array.

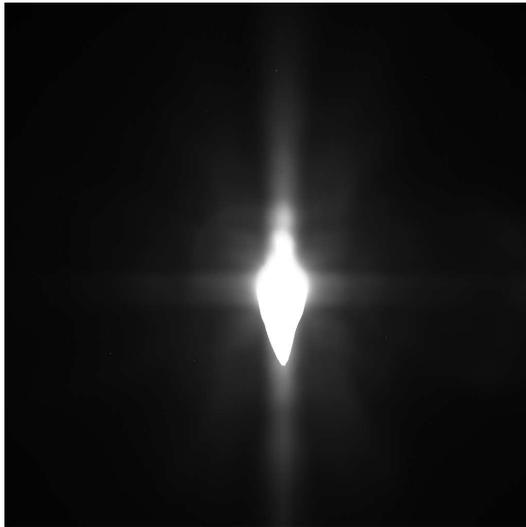
the same row (or column). This phenomenon, known as blooming, may cause bright streaks in a CCD image. A detailed discussion of blooming in CCDs may be found in [92]. Some CCDs include anti-blooming drains to reduce the impact of this problem, but this typically reduces the sensitivity of the array. Examples of blooming from real spacecraft images are shown in Fig 3.9. The image on the left was taken by the MESSENGER wide angle camera. In this image, the exposure time was too long and Earth completely saturates a large number of pixels in the array. The image on the right was taken by one of the star trackers on the Clementine spacecraft. In this image, the Moon is illuminated by Earth shine and the Sun is behind the Moon. Stray light from the Sun is visible around portions of the lunar horizon. Venus, which appears as a very bright dot in the upper right, saturates a number of pixels, leading to a vertical streak in the image. Therefore, blooming in CCDs may create artifacts in the image that should be addressed by the image processing algorithms in autonomous navigation applications.

3.4.2 Complementary Metal Oxide Semiconductors (CMOS)

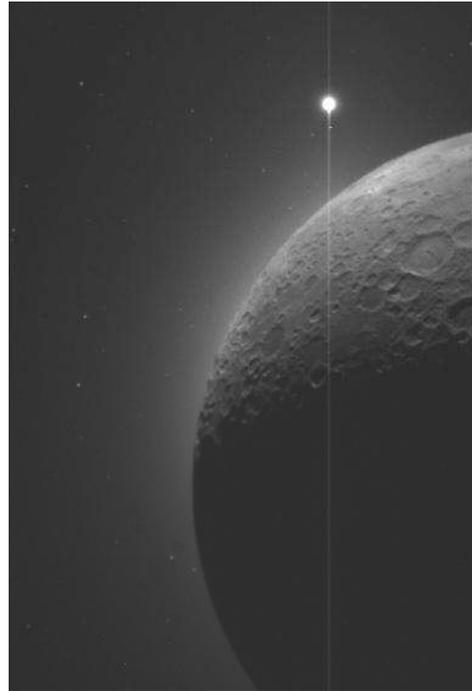
Complementary Metal Oxide Semiconductors (CMOS) sensors became competitive alternatives to CCD sensors in the 1990s. Both of these sensors use photoactive MOS capacitors to convert photons striking a pixel location into a stored charge packet. The primary difference between CCD sensors and CMOS sensors is how the charge packets are converted into a voltage. Recall that in CCDs, the charge packets for an entire row (or column) are transported

via a serial readout register to an output circuit where the conversion from charge to voltage takes place. In a CMOS sensor, however, the conversion from charge to voltage takes place separately in each pixel.

By converting the charge to voltage at each pixel, CMOS sensors do not experience blooming. This represents a natural advantage for CMOS sensors in many applications. Further, the readout architecture of CMOS sensors also allows for easy “windowing,” or the ability to selectively read out only a



a) Image of Earth taken by MESSENGER wide angle camera on 30 July 2005.



b) Image of the Moon and Venus taken by the Clementine Star Tracker B on 1 March 1994.

Figure 3.9: Examples of CCD blooming. Raw MESSENGER image is Product ID EW0031254402B from [93]. Raw Clementine image is Product ID LBA5876Z.193 from [94].

specific portion of the image. The method by which charges are transferred from gate to gate in CCD sensors makes windowing more difficult in CCDs.

These CMOS advantages, however, do come at a cost. The additional on-chip circuitry required to enable the CMOS readout architecture typically results in more noise. More detailed discussions about the differences between CCD sensors and CMOS sensors may be found in [95].

3.4.3 CCD/CMOS Quantum Efficiency and Fill Factor

Unfortunately, not every photon that hits a pixel will be converted into a photoelectron and be measured by the detector. First, the entire pixel surface is usually not covered with a photosensitive material. Therefore, define the fill factor, F , as the percentage of the pixel area that contains a photosensitive surface.

Further, there is not a 100% conversion for those photons that do strike the photosensitive surface. The quantum efficiency (QE) represents how efficient the photosensitive material is in converting photons to electrons. More specifically, QE is defined as the ratio of the number of photoelectrons generated to the number of photons that strike the surface (i.e. $QE = N_{pe}/N_{photons}$, where N_{pe} is the number of photoelectrons and $N_{photons}$ is the number of photons). Typically, QE is a function of the wavelength of the light. In the subsequent sections QE will be taken to be the average quantum efficiency over the spectral range of the detector.

Therefore, the number of photoelectrons generated by a pixel is given

by

$$N_{pe} = (QE)(F)(S)A_{pixel}\Delta t \quad (3.26)$$

where S is the total flux of photons (in units of photons/m²/sec) hitting the pixel, A_{pixel} is the total pixel area, and Δt is the exposure time over which photons are accumulated.

3.5 Overview of External Error Sources

External error sources are due to phenomena that are external to the optical sensor, and are determined by the environment in which the optical system is operating. The first two external error sources discussed are stellar aberration and parallax - both of these error sources affect the measured line-of-sight direction to an observed object. Fortunately, the procedures for correcting stellar aberration and parallax errors are well understood. The third external error source is stray light entering the optical system which causes additional photons to hit the detector and, therefore, causes increased noise in the image.

3.5.1 Stellar Aberration

Stellar aberration is a consequence of special relativity. This effect causes the line-of-sight unit vector to a star to look different in two inertial frames that have a relative velocity between them.

Begin by recalling that the line-of-sight to a star lies in the opposite direction of the velocity vector of the photons from that star. Now, suppose

the line-of-sight unit vector in some inertial frame (say a frame centered at the solar system barycenter) denoted by I is given by $(\mathbf{e}_i)_I$. Consider an additional frame, \mathcal{G} , whose axes are parallel to I , but whose origin is moving at a velocity \mathbf{v} relative to I . Due to the results of special relativity (and unlike what would be expected from Newtonian mechanics), $(\mathbf{e}_i)_I \neq (\mathbf{e}_i)_{\mathcal{G}}$.

The mathematical model presented here for stellar aberration is based on the elegant explanation provided by Shuster in 2003.[96] Only the final results are summarized here. If the variable $\boldsymbol{\beta}$ is defined as the velocity vector of the spacecraft normalized by the speed of light,

$$\boldsymbol{\beta} = \mathbf{v}/c \tag{3.27}$$

where c is the speed of light, then the rotation vector defining the stellar aberration error (to first order) is given by

$$\delta\boldsymbol{\theta}_{StellAberr} \approx \boldsymbol{\beta} \times (\mathbf{e}_i)_I \tag{3.28}$$

Therefore,

$$\mathbf{T}_{StellAberr} \approx \mathbf{I}_{3 \times 3} - [\delta\boldsymbol{\theta}_{StellAberr} \times] \tag{3.29}$$

$$(\mathbf{e}_i)_{\mathcal{G}} = \mathbf{T}_{StellAberr} (\mathbf{e}_i)_I \tag{3.30}$$

It is important to stress that there is not actually a rotation between frame I and \mathcal{G} ; the axes of these frames are parallel. There is only an apparent rotation in the line-of-sight direction to an object as seen in the two different frames due to the relative velocity between the frames.

To gain a feel for the magnitude of this error source, consider a spacecraft on a lunar return trajectory. As the spacecraft approaches Earth, its Earth-relative velocity may approach 10 km/s. Additionally assume that the reference directions to the stars are given in a frame attached to the solar system barycenter. Therefore, to arrive at a worst-case scenario, add the 29.8 km/s orbital velocity of the Earth to the Earth-relative spacecraft velocity, yielding a spacecraft heliocentric velocity of about 40 km/s. Therefore, applying Eq. 3.28, the magnitude of $\delta\theta_{StellAberr}$ may be as large as 27.5 arcsec.² This magnitude of error is not appreciably different from what would be expected for a LEO spacecraft.

Alternatively, consider a spacecraft on a Mars approach trajectory. Such a spacecraft is likely to have a Mars-relative velocity on the order of 6-9 km/s as it approaches periapsis on its hyperbolic approach trajectory (or just prior to Mars entry).[97] Because Mars has a heliocentric orbital velocity of about 24 km/s, a spacecraft could see a heliocentric velocity on the order of about 30 km/s. Using the same approach as above, this could yield a stellar aberration error as large as 20.6 arcsec.

Assuming that a reasonable estimate of the spacecraft heliocentric velocity is known, the measurement errors associated with stellar aberration may be removed. The residual error from this source after this correction is likely to be negligible for most reasonable scenarios.

²This result is similar in magnitude to that found by Shuster for a spacecraft in LEO.[96]

3.5.2 Parallax

Parallax is the apparent movement of the direction to a fixed target object due to a change in the location of the observer. This is shown graphically in Fig. 3.10. When objects are very far away, the parallax is very small and this effect is frequently neglected. Therefore, because the distance to most stars is very large, parallax is sometimes neglected for star observations. The reference directions found in a star catalog may be used directly, regardless of where the spacecraft is located, if parallax is ignored. In the present application, however, unmodeled parallax may present a small (but still noticeable) source of error for some star observations.

The following procedure for computing the error associated with unmodeled parallax closely parallels the approach used by Shuster.[96] If the position to a star in reference frame I is given by \mathbf{r}_{star} , then define the unit

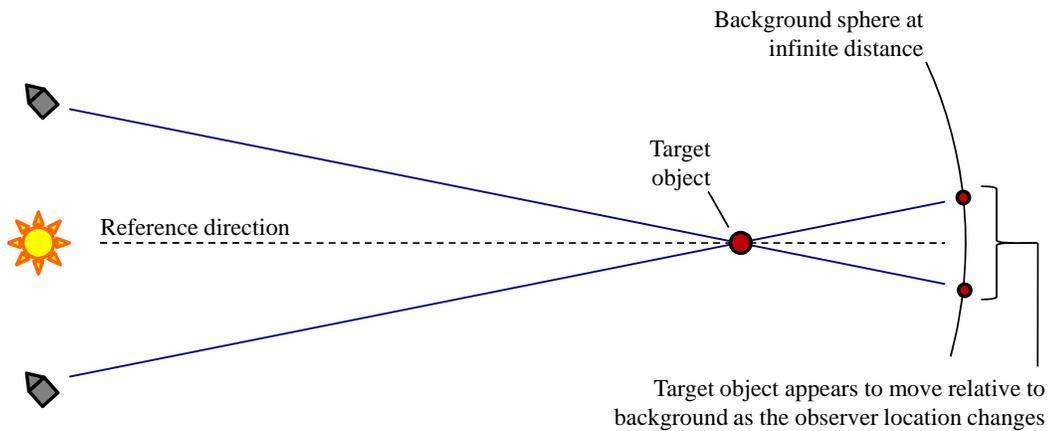


Figure 3.10: Graphical depiction of parallax. Angles are exaggerated for clarity.

vector to that star as $(\mathbf{e}_i)_I = \mathbf{r}_{star} / \|\mathbf{r}_{star}\|$. Additionally, suppose the position to the spacecraft (also in frame I) is given by \mathbf{r}_{sc} . If a new frame, I' , is defined with its axes parallel to I and centered at the spacecraft center of mass (i.e. displaced from I by \mathbf{r}_{sc}) then the line-of-sight vector from the spacecraft to the star in frame I' is given by

$$(\mathbf{e}_i)_{I'} = \frac{\mathbf{r}_{star} - \mathbf{r}_{sc}}{\|\mathbf{r}_{star} - \mathbf{r}_{sc}\|} = \frac{(\mathbf{e}_i)_I - \frac{\mathbf{r}_{sc}}{\|\mathbf{r}_{star}\|}}{\left\| (\mathbf{e}_i)_I - \frac{\mathbf{r}_{sc}}{\|\mathbf{r}_{star}\|} \right\|} \quad (3.31)$$

Taking the Taylor Series expansion and retaining only first order terms in $\mathbf{r}_{sc} / \|\mathbf{r}_{star}\|$ yields

$$(\mathbf{e}_i)_{I'} \approx (\mathbf{e}_i)_I + (\mathbf{e}_i)_I \times \left[(\mathbf{e}_i)_I \times \frac{\mathbf{r}_{sc}}{\|\mathbf{r}_{star}\|} \right] \quad (3.32)$$

Therefore, the rotation vector defining the parallax-induced error (to first order) is given by

$$\delta\boldsymbol{\theta}_{Parallax} \approx (\mathbf{e}_i)_I \times \frac{\mathbf{r}_{sc}}{\|\mathbf{r}_{star}\|} \quad (3.33)$$

where

$$\mathbf{T}_{Parallax} \approx \mathbf{I}_{3 \times 3} - [\delta\boldsymbol{\theta}_{Parallax} \times] \quad (3.34)$$

It is important to stress that there is not actually a rotation between frame I and I' . There is only an apparent rotation in the line-of-sight direction to an object fixed frame I as seen by an observer in frame I' .

To gain a feel for the magnitude of this error source, consider the same two examples discussed for stellar aberration. Additionally, suppose the observed star was Vega (apparent magnitude of 0.03 and brightest star in constellation Lyra) at a distance of 25 lightyears ($= 2.4 \times 10^{14}$ km) from the solar

system barycenter. For the case of the lunar return, where \mathbf{r}_{sc} is approximately equivalent to Earth's orbital radius, $\delta\theta_{Parallax}$ could be as large as 0.13 arcsec. For the case of the Mars approach, where \mathbf{r}_{sc} is approximately equivalent to Mars's orbital radius, $\delta\theta_{Parallax}$ could be as large as 0.2 arcsec.

Assuming that a reasonable estimate of the spacecraft heliocentric position is known, the measurement errors associated with parallax may be removed. As with stellar aberration, the residual error from parallax after the appropriate correction is made should be negligible for most reasonable scenarios.

3.5.3 Stray Light

Stray light is perhaps the most important external error source (assuming that the proper corrections are applied to remove the bulk of the error associated with stellar aberration and parallax). Stray light may be caused by a number of sources, including light that is reflected off of or scattered by:

1. Nearby objects such as planets, moons, asteroids, comets, or other nearby spacecraft
2. Dust and other particles in the camera FOV³
3. Thruster exhaust
4. Waste venting on a crewed vehicle

³The most prominent naturally occurring source of stray light from dust/particles is Zodiacal light. Zodiacal light is sunlight that is reflected off of the interplanetary dust cloud and is strongest in the ecliptic plane.

5. Nearby spacecraft surfaces
6. Any of the lenses/mirrors/filters in the optical system

The Sun is generally the primary source of stray light, although reflected sunlight from high albedo planets/moons can cause similar problems. Therefore, many star trackers, telescopes, and other spacecraft optical sensors have a Sun exclusion angle (minimum allowable angle between camera boresight and the Sun) that is significantly larger than what would be expected from simply considering direct sunlight and the camera FOV. For example, the Clementine star tracker (which had a $28^\circ \times 42^\circ$ FOV) required a Sun/Earth/Moon exclusion angle of $65^\circ \times 80^\circ$. [98] It should be noted, however, that planetary exclusion angles are not practical for optical navigation because the image must contain the planet. A Sun exclusion angle is still expected.

In addition to exclusion angles, there are many other techniques frequently used to control the amount of stray light that reaches the detector. The first, and perhaps most important, is a light baffle. The purpose of a light baffle system is to shield the optics from stray light originating outside of the intended FOV. Most modern light baffles consist of a tube (or cone) with a series of vanes used to attenuate the stray light. The purpose of each vane is to attenuate stray light directly hitting the vane or stray light that is scattered off the tip of another vane in the light baffle. In such a system, a stray light ray must bounce off of many surfaces before it is able to reach the detector. Because the surfaces in the light baffle are also coated to minimize the amount

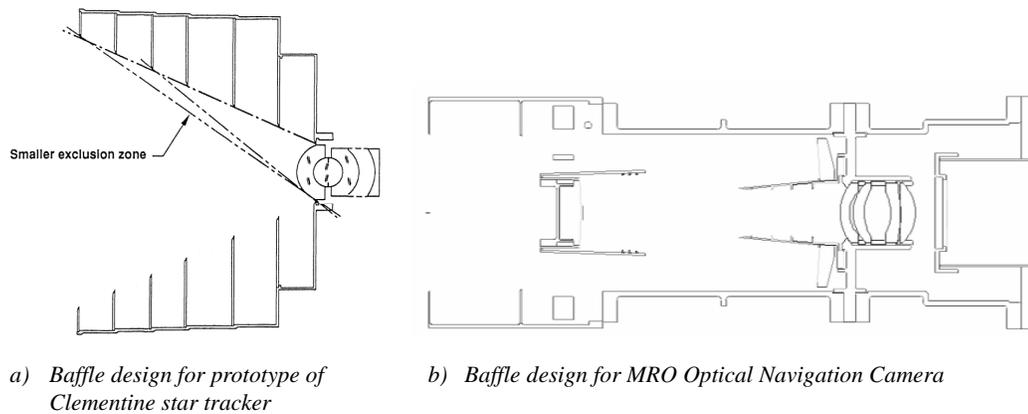


Figure 3.11: Examples of baffle design for the Clementine star tracker[101] and the Mars Reconnaissance Orbiter’s Optical Navigation Camera.[90] Both figures are reprinted with permission of SPIE and the authors.

of light that is reflected,⁴ the amount of stray light originating outside of the intended FOV that actually reaches the detector is very small. A well designed light baffle is capable of reducing the unwanted stray light flux by a factor of 10^{-5} to 10^{-9} . [100] Two examples of light baffle design are shown in Fig. 3.11.

Other useful techniques include stops (specifically aperture stops, field stops, and Lyot stops, depending on the optical design) and proper surface coatings of the lens/mirrors. Additionally, some systems may benefit from a filter to block stray light of a known wavelength. The optical navigation camera on MRO, for example, used a filter to block light with wavelengths above 650 nm.[90] This blocked the predominately red stray light from Mars while the spacecraft was imaging Phobos, Deimos, and background stars (as

⁴A common coating used for this purpose in space applications is Epner Laser Black.[99]

was shown in Fig. 2.14).

A more detailed discussion of practical methods for reducing stray light in spacecraft optical sensors may be found in [101], [102], and [90]. Subsequent analyses will assume that these techniques are applied to keep stray light levels below an acceptable level.

Regardless of what stray light reduction techniques are chosen, however, some amount of stray light will still reach the detector. Without modeling the details of the optics, light baffling, and external lighting environment,⁵ it is impossible to determine the structure and intensity of the stray light at each point on the detector. Because detailed optical design is not the purpose of the present analysis, it is assumed that the system is well designed and fabricated such that the primary source of stray light is from sunlight reflected off of dust particles in the camera FOV. It is further assumed that this stray light creates a uniform increase in the observed irradiance, given by $S_{SL,0}$ (in units of photons/m²/sec). Therefore, the total number of photons from stray light entering the camera system per unit time, Φ_{SL} , is given by

$$\Phi_{SL} = S_{SL,0} \frac{\pi}{4} d_{ap}^2 \mathbf{e}_{bs}^T \mathbf{e}_i \quad (3.35)$$

where d_{ap}^2 is the aperture diameter, $\mathbf{e}_{bs}^T \mathbf{e}_i = \cos\theta$, and θ is the angle between the boresight direction and the i -th line-of-sight direction. The $\cos\theta$ term is necessary to account for the apparent foreshortening of the aperture from a

⁵There are numerous commercially available optical design software packages capable of performing stray light analysis, such as ZEMAX and ASAP.

light source not along the boresight direction. It is worth noting that the FOV for most cameras is narrow enough to make this effect small for most applications. The lens will focus all of the photons coming from the direction \mathbf{e}_i onto (approximately) a single point on the focal plane. The irradiance at each pixel location may then be approximated as

$$S_{SL} = T S_{SL,0} \frac{\pi}{4} \frac{d_{ap}^2}{A_{pixel}} \mathbf{e}_{bs}^T \mathbf{e}_i \quad (3.36)$$

where T is the transmittance of the optics. Substituting these results into Eq. 3.26, the number of photoelectrons generated at each pixel by stray light, N_{SL} , is given by,

$$N_{SL} = (QE)(F) T S_{SL,0} \frac{\pi}{4} d_{ap}^2 \mathbf{e}_{bs}^T \mathbf{e}_i \Delta t \quad (3.37)$$

The stray light model chosen here means that the main effect of stray light in this simulation will be to increase the background noise in which the system must operate.

3.6 Overview of Internal Camera Error Sources: Alignment & Optics

3.6.1 Camera Mounting Misalignment

Limitations in the ability to mount the camera exactly as desired, coupled with the inability to measure the true orientation, will result in a camera mounting misalignment. These mounting errors will lead to errors in the estimate of \mathbf{T}_B^C in Eq. 3.23. Therefore, Eq. 3.23 may be rewritten to account for these errors,

$$(\mathbf{e}_i)_B = \mathbf{T}_B^C (\mathbf{e}_i)_C = \hat{\mathbf{T}}_B^C \mathbf{T}_{ma} (\mathbf{e}_i)_C \quad (3.38)$$

where \mathbf{T}_{ma} describes the rotation from the true camera frame to the estimate of the camera frame. Assuming that the misalignment error is small, the misalignment matrix is given by,

$$\mathbf{T}_{ma} = \mathbf{I}_{3 \times 3} - [\delta\boldsymbol{\theta}_{ma} \times] \quad (3.39)$$

3.6.2 Detector Plane Misalignment

The detector plane that contains the CCD/CMOS sensor array (x - y plane in frame D) may not lie exactly perpendicular to the optical axis. Additionally, the intersection of the true detector plane with the optical axis may be offset from the ideal detector frame described X - Y plane in frame C . These imperfections may lead to distortion of the image observed by the sensor. To develop a general 3-D model for the error introduced by detector plane misalignment, assume that the detector plane is tilted by a small angle α about the X -axis and a small angle β about the Y -axis. Note that there is no rotation about the Z -axis because of the choice of the camera reference frame discussed earlier. This leads to the following small angle rotation matrix for the detector plane misalignment, \mathbf{T}_{det} ,

$$\mathbf{T}_{det} = \mathbf{I}_{3 \times 3} - [\Delta\boldsymbol{\theta} \times] = \begin{bmatrix} 1 & 0 & -\beta \\ 0 & 1 & \alpha \\ \beta & -\alpha & 1 \end{bmatrix} \quad (3.40)$$

Therefore, the rotated plane that describes the orientation of the detector is defined by an x -axis of $\mathbf{i}_{rot} = [1 \ 0 \ -\beta]^T$ and a y -axis of $\mathbf{j}_{rot} = [0 \ 1 \ \alpha]^T$.

Additionally assume that there is an error in the separation of the lens with the detector along the optical axis given by δZ_l . Therefore, any point on the rotated plane may be described by

$$\mathbf{p} = \begin{bmatrix} 1 \\ 0 \\ -\beta \end{bmatrix} x_i + \begin{bmatrix} 0 \\ 1 \\ \alpha \end{bmatrix} y_i - \begin{bmatrix} 0 \\ 0 \\ \delta Z_l \end{bmatrix} \quad (3.41)$$

The ray that defines the path of the photon through the camera may be parameterized in terms of ξ by

$$\mathbf{p} = \begin{bmatrix} X_i \\ Y_i \\ 0 \end{bmatrix} + \begin{bmatrix} -X_i \\ -Y_i \\ Z_l \end{bmatrix} \xi \quad (3.42)$$

Setting these equations equal to each other and writing in matrix form,

$$\begin{bmatrix} X_i \\ Y_i \\ \delta Z_l \end{bmatrix} = \begin{bmatrix} X_i & 1 & 0 \\ Y_i & 0 & 1 \\ -Z_l & -\beta & \alpha \end{bmatrix} \begin{bmatrix} \xi \\ x_i \\ y_i \end{bmatrix} \quad (3.43)$$

An equation for ξ in terms of $[x_i, y_i]$ may be determined from the last row in the above matrix equation. Alternatively, an expression for ξ in terms of $[X_i, Y_i]$ may be determined from the analytic inversion of the above 3×3 matrix. Performing the required algebra will yield

$$\xi = \frac{-\beta x_i + \alpha y_i - \delta Z_l}{Z_l} = \frac{-\beta X_i + \alpha Y_i - \delta Z_l}{Z_l - \beta X_i + \alpha Y_i} \quad (3.44)$$

From here, a simple rearrangement of the first two rows in Eq. 3.43 yields the following expressions for $[x_i, y_i]$ in terms of $[X_i, Y_i]$:

$$x_i = (1 - \xi) X_i = \left(1 - \frac{-\beta X_i + \alpha Y_i - \delta Z_l}{Z_l - \beta X_i + \alpha Y_i} \right) X_i \quad (3.45)$$

$$y_i = (1 - \xi) Y_i = \left(1 - \frac{-\beta Y_i + \alpha Y_i - \delta Z_l}{Z_l - \beta X_i + \alpha Y_i} \right) Y_i \quad (3.46)$$

Arriving at the transform that goes the other way (i.e. $[X_i, Y_i]$ in terms of $[x_i, y_i]$) is also straightforward by using the definition of ξ in terms of x and y ,

$$X_i = \frac{1}{1 - \xi} x_i = \frac{Z_l}{Z_l + \beta x_i - \alpha y_i + \delta Z_l} x_i \quad (3.47)$$

$$Y_i = \frac{1}{1 - \xi} y_i = \frac{Z_l}{Z_l + \beta x_i - \alpha y_i + \delta Z_l} y_i \quad (3.48)$$

Because α , β , and δZ_l are small, the denominator in Eq. 3.47 and Eq. 3.48 may be expanded as a Taylor series and only first order terms retained. Rewriting this in matrix form,

$$\begin{bmatrix} X_i \\ Y_i \\ Z_l \end{bmatrix} \approx \mathbf{\Pi}_{det} \begin{bmatrix} x_i \\ y_i \\ Z_l \end{bmatrix} \quad (3.49)$$

where the matrix $\mathbf{\Pi}_{det}$ is given by

$$\mathbf{\Pi}_{det} = \begin{bmatrix} 1 - \frac{\beta x_i + \delta Z_l}{Z_l} & \frac{\alpha}{Z_l} x_i & 0 \\ -\frac{\beta}{Z_l} y_i & 1 + \frac{\alpha y_i - \delta Z_l}{Z_l} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.50)$$

Note that if α , β , and δZ_l go to zero, $\mathbf{\Pi}_{det}$ becomes the 3×3 identity matrix.

As a simple verification of this result, consider the 2D analog of the above 3D problem. The geometry of such a scenario is as shown in Fig. 3.12.

Note the following from the law of cosines:

$$X^2 = Z_l^2 + l^2 - 2Z_l l \cos\theta = l^2 - Z_l^2 \quad (3.51)$$

Additionally, note from Fig. 3.12 that $x = X + \delta$. Therefore, from Eq. 3.45, one would expect to see from this analysis that $\delta = -X\xi$. Using the law of

cosines a second time,

$$(X + \delta)^2 = (Z_l + \delta Z_l)^2 + (l - l\xi)^2 - 2(Z_l + \delta Z_l)(l - l\xi)\cos\theta \quad (3.52)$$

Note the sign on the $-l\xi$ term. This is negative because of the convention chosen in Eq. 3.42. Recognizing that δ , δZ_l , and ξ are small and keeping only first-order terms,

$$\begin{aligned} X^2 + 2X\delta &= Z_l^2 + 2Z_l\delta Z_l + l^2 - 2l^2\xi \\ &\quad - 2Z_l l \cos\theta + 2Z_l l \xi \cos\theta - 2\delta Z_l l \cos\theta \end{aligned} \quad (3.53)$$

and using the identity found in Eq. 3.51,

$$X\delta = Z_l\delta Z_l - l^2\xi + Z_l l \xi \cos\theta - \delta Z_l l \cos\theta \quad (3.54)$$

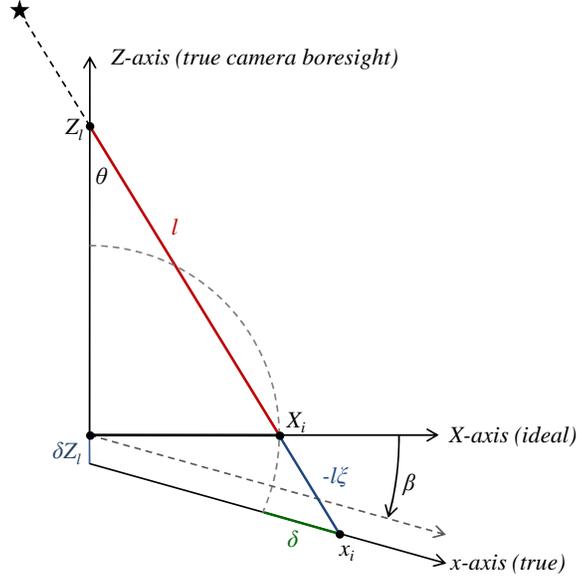


Figure 3.12: Geometry of camera detector plane misalignment.

Recognizing that $l\cos\theta = Z_l$ and once again using the identity in Eq. 3.51, it is straightforward to show that the 2D case confirms that to first order

$$\delta = -X\xi \quad (3.55)$$

3.6.3 Optical Aberrations

Optical aberrations are well understood phenomena that are described thoroughly in numerous texts.[88, 103] The third-order optical aberrations include: (1) spherical aberration, (2) coma, (3) astigmatism, and (4) distortion. The first three affect the quality of the image by blurring a point source. In a well designed optical system these errors are expected to be very small. More importantly, the size of these aberrations for a well designed optical system is expected to be smaller than a pixel, allowing these effects to be neglected in this application.

Distortion, on the other hand, shifts the location of a point source on the focal plane without affecting the quality of the image. Because the present work is primarily interested in determining the direction to an observed object, this is an important effect to consider. As will be shown later, filtering techniques allow object features to be located with subpixel accuracy. On this scale, radial distortion may not be negligible.

Radial distortion is most pronounced in cameras with a wide FOV. As the name implies, the magnitude of the distortion is a function of the radial distance of the observed point from the detector principal point. Fig. 3.4 shows that objects with a large angle between the object and the camera optical axis

project to a point significantly away from the principal point - these objects experience a larger radial distortion, as seen in Fig. 3.13. A simple radial distortion model is given by[86, 104]

$$X = X_d (1 + \kappa_1 r_d^2 + \kappa_2 r_d^4 + \kappa_3 r_d^6) \quad (3.56)$$

$$Y = Y_d (1 + \kappa_1 r_d^2 + \kappa_2 r_d^4 + \kappa_3 r_d^6)$$

where $[X, Y]$ are the undistorted coordinates, $[X_d, Y_d]$ are the distorted coordinates, and $r_d^2 = X_d^2 + Y_d^2$. Note that the radial distortion is defined in the X - Y plane of C . The coefficients κ_1 , κ_2 , and κ_3 are camera parameters that described the severity of the radial distortion. These coefficients are typically very small in magnitude.

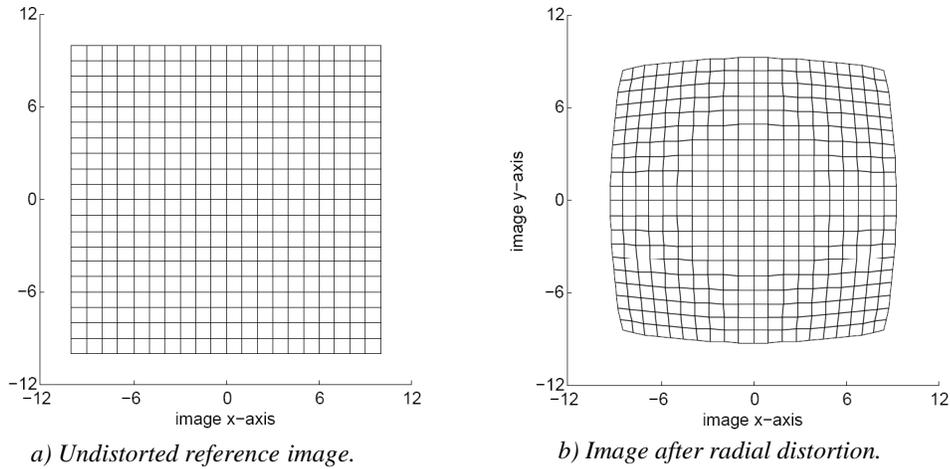


Figure 3.13: Effect of radial distortion on image.

3.6.4 Diffraction-Limited Spot Size in a Focused Image

Even if all imperfections in the optics can be removed (or be made negligibly small), there is still a theoretical minimum spot diameter for the image of a point source. A system that can achieve spot sizes on this scale is said to be “diffraction limited.” When a diffraction-limited optical system with a circular aperture is focused, the resulting spot is called an Airy disk. The empirically derived Rayleigh criterion is given by [88]

$$\sin\theta = 1.22 \frac{\lambda}{d_{ap}} \quad (3.57)$$

where θ is the angle between the point source and the first minimum of the Airy pattern, λ is the wavelength, and d_{ap} is the aperture diameter. Assuming that the angle θ is small (a good approximation because the focal length is much larger than the Airy disk), the diameter of the Airy disk is given by

$$d_{airy} = 2.44 \frac{\lambda f}{d_{ap}} \quad (3.58)$$

where f is the focal length and d_{ap} is the aperture diameter. Consider a camera with an f-number of F/1.43 (a focal length of 53 mm and an aperture diameter of 37 mm). Even if the star detected was on the edge of the visible spectrum ($\lambda = 750nm$), the diameter of the Airy disk would only be about $2.6\mu m$, an order of magnitude smaller than typical CCD/CMOS pixel widths. Therefore, accurate determination of the location of a point source on the image plane is expected to be limited by the detector and not the optics.

3.6.5 Point-Source Errors in a Defocused Image

Let the distribution of intensity for a given defocused object be defined by a point spread function (PSF). If the PSF of a defocused point source is assumed to have a bivariate Gaussian distribution and the blur along the u -axis and v -axis are uncorrelated and of the same magnitude, then the PSF is given by

$$S(u, v) = \frac{S_0}{2\pi\sigma^2} \exp \left\{ -\frac{(u - u_0)^2 + (v - v_0)^2}{2\sigma^2} \right\} \quad (3.59)$$

where S_0 is the intensity of the point source and $[u_0, v_0]$ is the centroid of the point source as described in frame A . There is extensive precedence in the literature for using a Gaussian PSF to model the blurring associated with camera defocus.[105–107]

3.7 Overview of Internal Camera Error Sources: CCD and CMOS Detectors

The variance of the total noise associated with the intensity measurement at each pixel is given by

$$\sigma_{sys}^2 = \sigma_{shot}^2 + \sigma_{floor}^2 + \sigma_{reset}^2 + \sigma_{pattern}^2 + \sigma_{ADC}^2 \quad (3.60)$$

where σ_{sys}^2 is the variance of the total system noise, σ_{shot}^2 is the variance of the shot noise, σ_{floor}^2 is the variance of the noise floor, σ_{reset}^2 is the variance of the reset noise, $\sigma_{pattern}^2$ is the variance of the pattern noise, and σ_{ADC}^2 is the variance of the analog to digital conversion (ADC) noise. The typical net result of such noise sources in a dark frame is as shown in Fig. 3.14. Each of these

noise terms shown in Eq. 3.60 is discussed in more detail in the subsequent sections.

3.7.1 Dark Current and Shot Noise

Even when no photons are striking the photosensitive surface, some electrons are still thermally generated. Electrons generated through this process are referred to as dark current.[95, 107] To estimate the mean dark current in real-time, many detectors have a set of dark pixels that are shielded from external lighting. The only signal generated from these dark pixels is from dark current. The signal from these pixels is averaged and may be subtracted from the active pixels to remove the mean dark current effect.

Dark current is a strong function of detector temperature and the am-

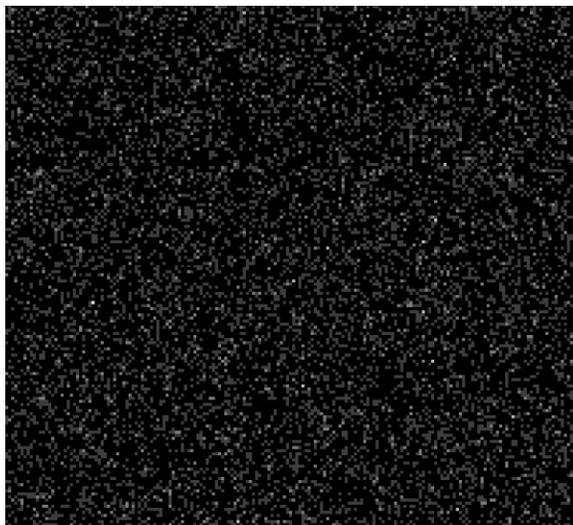


Figure 3.14: Background noise on a dark frame (contrast significantly increased to make noise more visible).

bient radiation environment. The dark current density generated by a pixel is given by[92]

$$J_D = aT^{1.5} \exp \left[\frac{-E_G}{2kT} \right] \quad (3.61)$$

where k is the Boltzmann constant, T is temperature in Kelvin, and E_G is the band gap energy. The constant a has units of A/K^{1.5}/m² and is a constant for the detector of interest, resulting in a dark current density with units of A/m². The band gap energy is also a function of temperature. The following empirical relationship is given in [92] as a good approximation for this quantity:

$$E_G = 1.557 - \frac{(7.021 \times 10^{-4}) T^2}{1108 + T} \quad (3.62)$$

where the resulting E_G is in eV and the T is the temperature in Kelvin. Therefore, the number of dark current generated electrons is given by

$$N_{dark} = \frac{J_D A_{pixel} \Delta t}{q} \quad (3.63)$$

where A_{pixel} is the area of the entire pixel (note that the entire pixel, not just the photosensitive surface, generates dark current) and $q = 1.6 \times 10^{-19}$ coulombs/e⁻ is the charge of an electron.

Electrons generated from photons striking the detector (which come from the observed object and stray light) and electrons generated from dark current both contribute to the mean value of electrons detected by a given pixel:

$$E [N_{sys}] = N_{pe} + N_{SL} + N_{dark} \quad (3.64)$$

where N_{sys} is the total number of electrons generated at a particular location, N_{pe} is the number of photoelectrons generated by light from the observed object, N_{SL} is the number of photoelectrons generated by stray light, and N_{dark} is the number of electrons generated by dark current. The noise associated with the generation of electrons (which must come in discrete values) is frequently called shot noise. Shot noise is well known to follow a Poisson distribution, so the variance of the shot noise is equivalent to the mean number of electrons. Therefore,

$$\sigma_{shot}^2 = N_{pe} + N_{SL} + N_{dark} \quad (3.65)$$

For large values of $(N_{pe} + N_{SL} + N_{dark})$, the Poisson distribution looks like a Gaussian distribution.

3.7.2 Amplifier Noise

The CCD/CMOS output amplifier is also a source of noise. Output amplifier noise comes from flicker noise (sometimes called 1/f noise) and white noise. Because flicker noise is inversely proportional to frequency (or pixel rate), the flicker noise may be very low compared to the white noise for detectors operating at a sufficiently large pixel rate. The term “pixel rate” describes the rate at which pixels are read from the camera readout register. As the frequency is increased, the RMS value of the output amplifier noise will approach the white noise. This minimum noise level is called the noise floor.

Recall that the pixel well size is defined by the number of photoelectrons required to saturate the pixel. From here, it is easy to see that the noise

floor and the pixel well size must be related by the detector dynamic range. Therefore, the noise floor is given by

$$\sigma_{floor} = N_{max}/DR \quad (3.66)$$

where N_{max} is the pixel well size and DR is the sensor dynamic range. Note that DR is in linear units. Manufacturers frequently report dynamic range in decibels (i.e. $DR_{dB} = 20 \log_{10} DR$).

3.7.3 Reset Noise

Reset noise is generated when the sense node capacitor on the pixel is reset. Although becoming less of a problem on modern sensors, the reset noise has historically been significant noise source in CMOS detectors.[108] Sometimes called kTC noise, the reset noise is given by [95]

$$\sigma_{reset} = \frac{\sqrt{kTC}}{q} \quad (3.67)$$

where k is the Boltzmann constant, T is the temperature, C is the sense node capacitance, and $q = 1.6 \times 10^{-19}$ col/e⁻ is the charge of an electron. The manufacturer frequently gives the reset noise at some specified reference temperature. Therefore,

$$\sigma_{reset} = \sigma_{reset,ref} \sqrt{\frac{T}{T_{ref}}} \quad (3.68)$$

3.7.4 Pattern Noise

There are two contributing sources to pattern noise. The first is Fixed Pattern Noise (FPN) that describes pixel-to-pixel variation in the signal when the detector is in the dark. This phenomenon is primarily the result of pixel-to-pixel differences in dark current. Note that this variation is always present (or “fixed”), regardless of whether any photons are striking the photosensitive surface. The FPN is usually described as a percentage of the number of dark current electrons. The second term in the pattern noise is photo-response non-uniformity (PRNU) noise. This noise term describes the pixel-to-pixel variation in sensitivity to photons. Therefore, PRNU is usually described as a percentage of the number of photoelectrons generated by the pixel.

Recalling that the variance of the number of photoelectrons and the number of dark current electrons are both governed by a Poisson distribution, the total variance for pattern noise is given by

$$\sigma_{pattern}^2 = (U_{fpm}N_{dark})^2 + (U_{prnu}N_{pe})^2 \quad (3.69)$$

3.7.5 ADC Quantization Noise

Most detectors report the intensity at each pixel in digital form instead of analog form. Therefore, there is some quantization associated with the analog to digital conversion. The digitized output is given by

$$N_{pe} = \alpha \text{ floor } \left[\frac{N_{pe}}{\alpha} \right] \quad (3.70)$$

where the ‘floor’ operator rounds down to the nearest integer and α is the least significant bit (LSB) and is given by

$$\alpha = \frac{N_{max}}{2^m} \quad (3.71)$$

where N_{max} is the pixel well capacity in electrons and m is the number of bits of the digital output. Because a simple rounding function is used, the ADC noise is best described by a uniform distribution. Therefore, recalling that the variance of a unit uniform distribution is $1/12$, the variance associated with this process is

$$\sigma_{ADC}^2 = \frac{\alpha^2}{12} \quad (3.72)$$

3.8 Camera Calibration

Many of the internal camera error sources associated with optics and alignment (discussed in Section 3.6) may be largely removed through proper camera calibration. These error sources can usually be estimated well enough that they may be ignored after the corrections described in previous sections are applied. A detailed discussion of practical camera calibration techniques may be found in [109], [110], and [111].⁶ As an example, consider the calibration approach used for the Cassini cameras.[52] Let the measured coordinates

⁶An additional resource of particular note is the “Camera Calibration Toolbox for MATLAB” developed by the Computer Vision Research Group at the California Institute of Technology. This toolbox is available electronically at http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

of the observed object, $[\tilde{x}, \tilde{y}]$, be related to the true object coordinates, $[x, y]$, by

$$x = \tilde{x} + \delta x \quad y = \tilde{y} + \delta y \quad (3.73)$$

where δx and δy are the errors in the $[x, y]$ coordinates due to distortion and misalignment of internal components. The Cassini program modeled these errors through the following camera calibration model,

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \begin{bmatrix} -\tilde{y} \tilde{r} & \tilde{x} \tilde{r}^2 & -\tilde{y} \tilde{r}^3 & \tilde{x} \tilde{r}^4 & \tilde{x} \tilde{y} & \tilde{x}^2 \\ \tilde{x} \tilde{r} & \tilde{y} \tilde{r}^2 & \tilde{x} \tilde{r}^3 & \tilde{y} \tilde{r}^4 & \tilde{y}^2 & \tilde{x} \tilde{y} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} \quad (3.74)$$

where $\tilde{r}^2 = \tilde{x}^2 + \tilde{y}^2$. A number of measurements of known objects were made and a batch estimation procedure was used to estimate the six calibration coefficients, a_i . For the Cassini mission, only the coefficients a_2 , a_5 , and a_6 were found to be important.[52] Looking at the physical camera models developed in Section 3.6, the source of these errors can easily be identified.

First, comparing Eq. 3.74 with Eq. 3.56, it is clear that the calibration coefficient a_2 corresponds to the quadratic radial distortion term ($a_2 = \kappa_1$) and the calibration coefficient a_4 corresponds to the quartic radial distortion term ($a_4 = \kappa_2$). Of note is that a_2 was the only important parameter related to distortion, meaning that the Cassini camera did not have a significant quartic radial distortion term.

The calibration coefficients a_5 and a_6 may be shown to approximately describe the detector plane misalignment. Looking at the misalignment matrix

$\mathbf{\Pi}_{det}$ from Eq. 3.50 and ignoring the error in the focal length,

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \approx \begin{bmatrix} 1 - \frac{\beta}{Z_l} \tilde{x} & \frac{\alpha}{Z_l} \tilde{x} & 0 \\ -\frac{\beta}{Z_l} \tilde{y} & 1 + \frac{\alpha}{Z_l} \tilde{y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} \quad (3.75)$$

Therefore,

$$\delta x \approx -\frac{\beta}{Z_l} \tilde{x}^2 + \frac{\alpha}{Z_l} \tilde{x} \tilde{y} \quad (3.76)$$

$$\delta y \approx -\frac{\beta}{Z_l} \tilde{x} \tilde{y} + \frac{\alpha}{Z_l} \tilde{y}^2 \quad (3.77)$$

which means that $a_5 \approx \alpha/Z_l$ and $a_6 \approx -\beta/Z_l$. If the focal length is known, then knowledge of the calibration coefficients a_5 and a_6 may be used to determine the angular misalignment of the detector (which is given by the small angle α about the X -axis and the small angle β about the Y -axis).

3.9 Light Intensity Models for Observed Objects

Two distinct types of objects are expected to be observed in simulated OPNAV images. The first type of object is a star, which appears as a point source of light in a focused image. The second type of object is a nearby celestial body (such as a planet or moon), which will illuminate many pixels in the image. What follows is a discussion for the simulation of these two types of objects.

3.9.1 Simulation of Stars

If a star is in the camera FOV, begin by applying the parallax and stellar aberration corrections to the catalog reference directions. Next, transform the

inertial unit vector to the camera frame,

$$(\mathbf{e}_i)_C = \mathbf{T}_{ma}^T \mathbf{T}_C^B \mathbf{T}_B^I \mathbf{T}_{StellAberr} \mathbf{T}_{Parallax} (\mathbf{e}_i)_I \quad (3.78)$$

From here, proceed to the computation of the ideal X - Y coordinates,

$$X_i = -\frac{(\mathbf{e}_i(1))_C}{(\mathbf{e}_i(3))_C} Z_l \quad Y_i = -\frac{(\mathbf{e}_i(2))_C}{(\mathbf{e}_i(3))_C} Z_l \quad (3.79)$$

Next distort the image using Eq. 3.56 to get X_d and Y_d (this equation must be solved backwards with Newton-Raphson iteration). Using the distorted X - Y coordinates, compute the measured x - y centroid coordinates on the detector plane using Eq. 3.45 and Eq. 3.46. Finally, use Eq. 3.16 to convert the x - y detector coordinates to the u - v array coordinates.

Once the centroid has been computed, the intensity of the source must be determined. For a star, a simple model for the photon flux is given by[88]

$$\Phi = S_{0\lambda} T \frac{\pi}{4} d_{ap}^2 \Delta\lambda 10^{-0.4m} \quad (3.80)$$

where $S_{0\lambda}$ is the spectral irradiance⁷ of a zero-magnitude star (about 10^7 photons·cm⁻²·s⁻¹·μm⁻¹), T is the transmittance of the optics, d_{ap} is the aperture diameter, $\Delta\lambda$ is the spectral range of interest, and m is the star visual magnitude. This simple model may be extended to account for starlight not originating from along the camera boresight direction, resulting in an irradiance at the detector plane of

$$S_i = S_{0\lambda} T \frac{\pi}{4} \frac{d_{ap}^2}{A_{pixel}} \Delta\lambda 10^{-0.4m} \mathbf{e}_{bs}^T \mathbf{e}_i \quad (3.81)$$

⁷Spectral irradiance is irradiance per unit wavelength.

where S_i is the irradiance from the i -th star at the detector plane, $\mathbf{e}_{bs}^T \mathbf{e}_i = \cos\theta$, and θ is the angle between the boresight direction and the i -th line-of-sight direction. As with the stray light model, the $\cos\theta$ term is necessary to account for the apparent foreshortening of the aperture from a light source not along the boresight direction. It is worth noting that the FOV for most cameras is narrow enough to make this effect small for most applications. The lens will focus all of the photons coming from the direction \mathbf{e}_i onto (approximately) a single point on the focal plane. Substituting these results into Eq. 3.26, the number of photoelectrons generated by the i -th star is given by,

$$N_i = (QE)(F) S_{0\lambda} T \frac{\pi}{4} d_{ap}^2 \Delta\lambda 10^{-0.4m} \mathbf{e}_{bs}^T \mathbf{e}_i \Delta t \quad (3.82)$$

With the centroid location and ideal intensity available, the Gaussian PSF is applied to spread the intensity to neighboring pixels. For a single point source, the intensity may be spread among pixels by at least two different methods. The first is a Monte Carlo approach where a large number of dispersed points are randomly created and the the intensity of each pixel is determined by the percentage of points that fall within that pixel. The second is to convolve a linear Gaussian filter with a perfectly focused version of the image. Use of a linear filter to model defocusing may also be nicely extended into images that includes things other than point sources (such as images that contain nearby planets, moons, or asteroids - a subject to be discussed more in the next section). The Monte Carlo approach was used to blur stars in most of the simulations performed in this dissertation.

3.9.2 Simulation of Planets and Moons

Simulating the appearance of an extended body, such as a planet or moon, in an image is a three step process. The first step is determining which pixels are illuminated by the body of interest. The second step is determining how light is reflected off of each observed point on the body of interest. The third step is determining the number of photons that strike each of the pixels on the detector illuminated by the body of interest. These steps are described in detail in the following subsections.

3.9.2.1 Step 1: Find Pixels Illuminated by Planet/Moon

The first step is to sweep through all of the pixel locations in the image and determine which pixels are illuminated by the body of interest. At each $[u, v]$ pixel coordinate, the ideal $[X, Y]$ coordinates in the camera frame may be constructed from Eq. 3.17, Eq. 3.47, and 3.48. The line-of-sight vector in the camera frame, $(\mathbf{e}_i)_C$, may now be computed from Eq. 3.22. Finally, $(\mathbf{e}_i)_C$ may be transformed into the planetcentric frame to obtain $(\mathbf{e}_i)_I$,

$$(\mathbf{e}_i)_I = \mathbf{T}_{StellAberr}^T \mathbf{T}_I^B \mathbf{T}_B^C \mathbf{T}_{ma} (\mathbf{e}_i)_C \quad (3.83)$$

Note that, unlike with stars, no parallax term is needed because the planet-spacecraft relative geometry is modeled explicitly.

The task is now to determine if a ray originating from the camera and extending in the direction of $(\mathbf{e}_i)_I$ intersects the planet or moon. Assuming that the planet or moon may be modeled by a triaxial ellipsoid, any point on

the surface of this planet or moon must satisfy

$$\mathbf{p}^T \begin{bmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/b^2 & 0 \\ 0 & 0 & 1/c^2 \end{bmatrix} \mathbf{p} = \mathbf{p}^T \mathbf{A} \mathbf{p} = 1 \quad (3.84)$$

where \mathbf{p} is the location of any surface point in a planetcentric frame. The values a , b , and c are the principal axis dimensions of the planet/moon. If \mathbf{r}_c is the position of the camera with respect to the planet center, then the vector from the camera to this surface point is given by

$$\mathbf{s} = \mathbf{p} - \mathbf{r}_c \quad (3.85)$$

This geometry may be seen in Fig. 3.15. The vector \mathbf{s} , of course, is along the same direction as $(\mathbf{e}_i)_I$, but with some unknown magnitude, t . Therefore

$$\mathbf{s} = \mathbf{p} - \mathbf{r}_c = t (\mathbf{e}_i)_I \quad (3.86)$$

Solving for \mathbf{p} and substituting the result into Eq. 3.84,

$$[t (\mathbf{e}_i)_I + \mathbf{r}_c]^T \mathbf{A} [t (\mathbf{e}_i)_I + \mathbf{r}_c] = 1 \quad (3.87)$$

Expanding the above yields a quadratic in the scale variable t ,

$$\left[(\mathbf{e}_i)_I^T \mathbf{A} (\mathbf{e}_i)_I \right] t^2 + \left[(\mathbf{e}_i)_I^T \mathbf{A} \mathbf{r}_c + \mathbf{r}_c^T \mathbf{A} (\mathbf{e}_i)_I \right] t + \left[\mathbf{r}_c^T \mathbf{A} \mathbf{r}_c - 1 \right] = 0 \quad (3.88)$$

The selected pixel will view the planet if a real root exists for t . Therefore, a pixel will be illuminated by the planet or moon if

$$\left[(\mathbf{e}_i)_I^T \mathbf{A} \mathbf{r}_c + \mathbf{r}_c^T \mathbf{A} (\mathbf{e}_i)_I \right]^2 - 4 \left[(\mathbf{e}_i)_I^T \mathbf{A} (\mathbf{e}_i)_I \right] \left[\mathbf{r}_c^T \mathbf{A} \mathbf{r}_c - 1 \right] \geq 0 \quad (3.89)$$

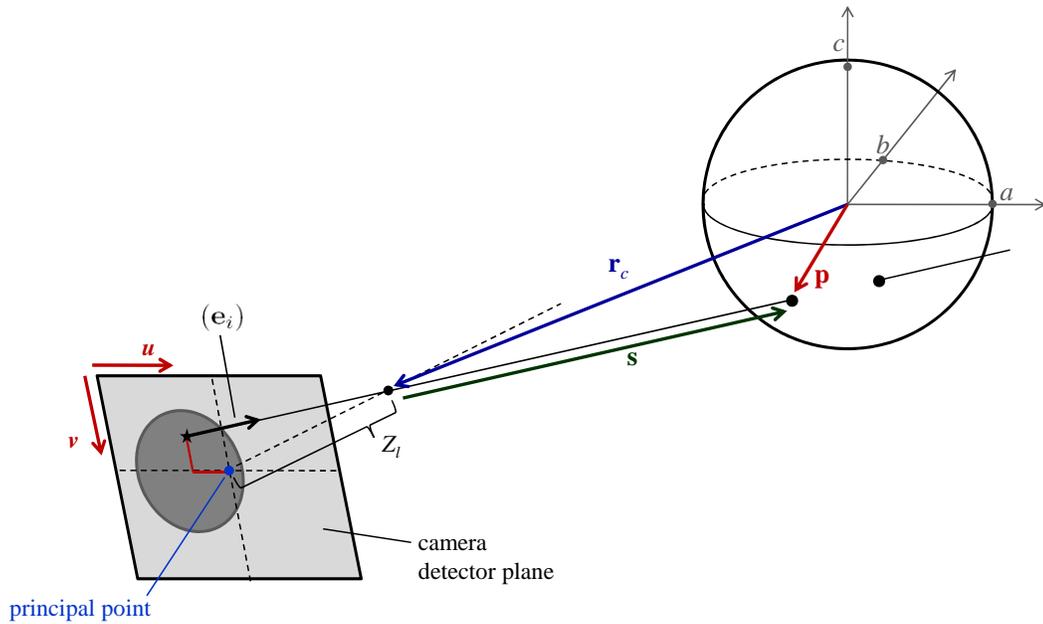


Figure 3.15: Geometry of image of planet or moon.

In general, two real roots will exist for any unit vector direction that intersects the planet/moon. A repeated root will occur if the point is on the observed limb of the planet/moon. Given the geometry in Fig. 3.15, the smaller root is always desired because only the point on the near side of the planet/moon is observed.

3.9.2.2 Step 2: Find the Intensity of Reflected Light

If a pixel is found to be illuminated by a planet or moon by the satisfaction of Eq. 3.89, then the second step is to determine the intensity of the light reflected off the patch of the planet/moon viewed by this pixel. It is assumed that light within our solar system originates from the Sun and when

a planet/moon is imaged the observed light is reflected sunlight.

The simplest way to model this reflected sunlight would be to assume Lambertian reflection. Such a model assumes that the light reflected is proportional to the cosine of the angle of incidence and completely independent of the location of the observer. This model was found to be insufficient for the present application.

Instead, bidirectional reflectance theory is used to model the sunlight reflected by a planet or moon.⁸ Define the bidirectional reflectance, r , as the ratio of the radiance⁹ of light reflected off a surface and viewed from a specific direction, I , to the irradiance incident on the surface, S_0 . The Lommel-Seeliger Law is a common, simple model used in planetary photometry. According to this relation,[114]

$$r(i, \theta) = \frac{w}{4\pi} \frac{\cos i}{\cos i + \cos \theta} \quad (3.90)$$

where w is the average single-scattering albedo, i is the angle of incidence, and θ is the angle of reflectance. In [112], Hapke introduces a more sophisticated model that expresses the bidirectional reflectance r as a function of three angles: the angle of incidence, i ; the angle of reflectance, θ ; and the phase angle, g (see Fig. 3.16 for geometry). The bidirectional reflectance is given by

$$r(i, \theta, g) = I/S_0 \quad (3.91)$$

⁸A detailed discussion of bidirectional reflectance theory may be found in [112] and [113].

⁹Radiance is radiant power per unit area per unit solid angle, or $\text{W} \cdot \text{m}^{-2} \cdot \text{sr}^{-1}$. Note the difference in “radiance” and “irradiance.” For clarity, radiance will be denoted by the variable I and irradiance will be denoted by the variable S .

$$r(i, \theta, g) = \frac{w}{4\pi} \frac{\cos i}{\cos i + \cos \theta} \{ [1 + B(g)] P(g) + H(\cos i) H(\cos \theta) - 1 \} \quad (3.92)$$

where $P(g)$ is the average phase angle function, $B(g)$ is the back-scattering function, and $H(\cdot)$ is defined as follows to compact notation

$$H(x) = \frac{1 + 2x}{1 + 2\gamma x} \quad (3.93)$$

where the variable γ is defined as,

$$\gamma = (1 - w)^{1/2} \quad (3.94)$$

The variables w and γ describe the albedo of the planet or moon. For an isotropic surface, the relationship between the bond albedo, A_b , and the average single-scattering albedo is known to be given by[112]

$$A_b \approx \frac{1 - \gamma}{1 + \gamma} \left[1 - \frac{1}{3} \frac{\gamma}{1 + \gamma} \right] \quad (3.95)$$

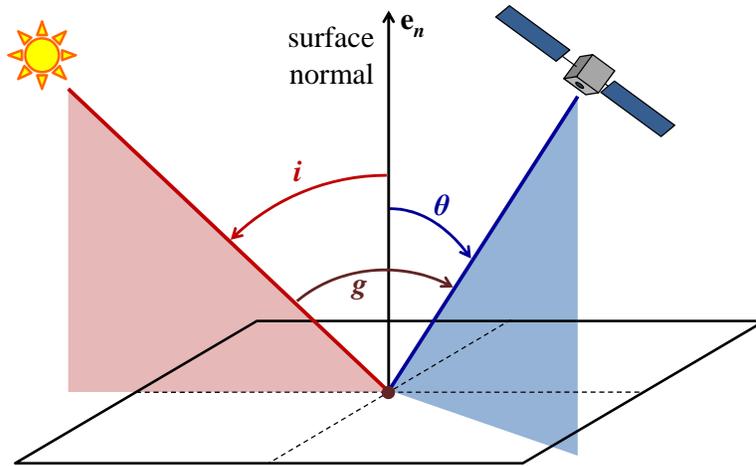


Figure 3.16: Geometry of bidirectional reflectance.

The bond albedo is defined as the light scattered in all directions by a surface to the light incident on that surface. Using Eq. 3.95, a plot of the bond albedo as a function of the average single-scattering albedo is shown in Fig. 3.17. These results may be cross-referenced against the average bond albedos (sometimes called the average planetary albedo) for the planets shown in Table 3.2.

Evaluation of Eq. 3.92 requires a model for the average phase angle function, $P(g)$. For isotropic scattering, $P(g) = 1$. Hapke demonstrated in the 1960s that the following is a good approximation of $P(g)$ for a lunar-like surface,[116, 117]

$$P(g) = \frac{4\pi}{5} \left[\frac{\sin g + (\pi - g)\cos g}{\pi} + \frac{(1 - \cos g)^2}{10} \right] \quad (3.96)$$

This model has since been used for many other planetary applications.[52, 112]

A model is also required for the back-scattering function, $B(g)$. For

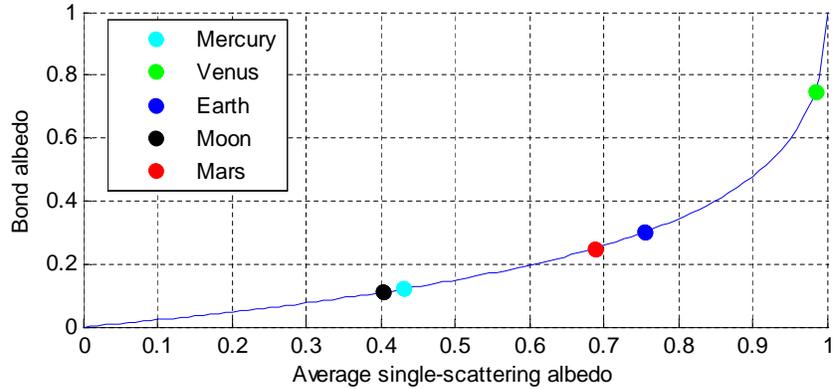


Figure 3.17: Relationship between average single-scattering albedo and average bond albedo.

Table 3.2: Average bond albedo for selected celestial bodies.[115]

Celestial body	Average albedo
Mercury	0.12
Venus	0.75
Earth	0.30
Moon	0.11
Mars	0.25
Jupiter	0.34
Saturn	0.34
Uranus	0.30
Neptune	0.29
Pluto	0.5

low albedo bodies, an observer should expect to see regions of illumination and regions of shadow based on the topography of the body. For high albedo bodies, however, the scattering of light from illuminated areas will cause regions not directly illuminated by sunlight (regions that would be in shadow for low albedo objects) to also be illuminated. This phenomena is called back-scattering. Direct sunlight, therefore, produces surface illumination dependent on the first power of albedo, while back-scattering produces surface illumination dependent on second power (and higher powers) of albedo. In addition to albedo, the amount of back-scattering is also dependent on the geometric properties of the surface that is reflecting the light (e.g. roughness, porosity, etc.). The back-scattering function, therefore, may be parameterized on an additional variable, h . If $\kappa = h / \tan |g|$, then the back-scattering function is given by

$$B(g) = \begin{cases} B_0 \left[1 - \frac{1}{2\kappa} (3 - e^{-\kappa}) (1 - e^{-\kappa}) \right] & |g| < \pi/2 \\ 0 & |g| \geq \pi/2 \end{cases} \quad (3.97)$$

where B_0 may be approximated by

$$B_0 = \exp(-w^2/2) \quad (3.98)$$

Values of h for Earth's moon, for example, vary between 0.4 and 0.8.[117]

All that remains to compute the bidirectional reflectance, r , from Eq. 3.92 is to find the angles i , θ , and g . These angles are referenced to the ellipsoidal planet's surface normal at the location where the light is reflected. Recall that the surface normal for such an object is simply

$$\mathbf{p}_n = \mathbf{A}\mathbf{p} \quad \mathbf{e}_n = \frac{\mathbf{P}_n}{\|\mathbf{P}_n\|} \quad (3.99)$$

Therefore, the angles required to complete Eq. 3.92, Eq. 3.96, and Eq. 3.97 are easily computed as

$$i = \text{acos}[\mathbf{e}_n^T \mathbf{e}_{sun}] \quad (3.100)$$

$$\theta = \text{acos}[-\mathbf{e}_n^T (\mathbf{e}_i)_I] \quad (3.101)$$

$$g = \text{acos}[-\mathbf{e}_{sun}^T (\mathbf{e}_i)_I] \quad (3.102)$$

where \mathbf{e}_{sun} is a unit vector pointing from the center of the planet/moon to the Sun.

With the bidirectional reflectance known, the radiance of the reflected light, I , may be found by

$$I = r S_0 \quad (3.103)$$

While r may vary with wavelength (because w may vary with wavelength), it is assumed that the process of reflection does not shift the wavelength of the

incident light. The central issue moving forward is the determination of S_0 (the irradiance of the light incident on the surface). If the incident irradiance is sunlight, the following simple model may be used

$$S_0 = S_{ref} \left(\frac{r_{ref}}{\|\mathbf{r}_{sc}\|} \right)^2 \quad (3.104)$$

where \mathbf{r}_{sc} is the position vector of the spacecraft with respect to the Sun and S_{ref} is the reference irradiance of the Sun at a distance of r_{ref} from the Sun. The value of S_{ref} to be used in this application is dependent on the spectral range optical sensor. Sunlight at a frequency outside of the spectral range of the CCD/CMOS sensor or outside of the bandpass region of the optical filter will not be absorbed by the detector and need not be considered.

It is well known that the spectrum of the Sun is very close to that of a black body with an effective temperature of 5777 K.[115] Therefore, recalling Planck's law as a function of wavelength, λ , the solar spectral irradiance ($\text{W} \cdot \text{m}^{-2} \cdot \text{m}^{-1}$) for a segment of the Sun's surface is[115, 118]

$$S_\lambda(\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \left[\exp\left(\frac{hc}{\lambda kT}\right) - 1 \right]^{-1} \quad (3.105)$$

where h is Planck's constant, c is the speed of light, k is the Boltzmann constant, and T is the effective temperature. This may easily be converted to the solar spectral irradiance at a reference distance,

$$S_{ref,\lambda} = \frac{2\pi hc^2}{\lambda^5} \left[\exp\left(\frac{hc}{\lambda kT}\right) - 1 \right]^{-1} \left(\frac{r_{Sun}}{r_{ref}} \right)^2 \quad (3.106)$$

where r_{Sun} is the radius of the Sun. The solar irradiance over the spectral

range from λ_{min} to λ_{max} , therefore, is given by

$$S_{ref} = \int_{\lambda_{min}}^{\lambda_{max}} S_{ref,\lambda} d\lambda \quad (3.107)$$

In many cases, it is more important to know the solar irradiance in terms of photons $\cdot \text{m}^{-2} \cdot \text{s}^{-1}$ instead of $\text{W} \cdot \text{m}^{-2}$. Recalling that the energy of a photon is given by

$$E_{photon} = \frac{hc}{\lambda} \quad (3.108)$$

it is straightforward to show that

$$S_{ref} = \int_{\lambda_{min}}^{\lambda_{max}} \frac{\lambda}{hc} S_{ref,\lambda} d\lambda \quad (3.109)$$

The results of this section may now be combined to provide a tractable approach for computing the radiance of the light reflected from an observed surface patch on a planet or moon (which was the objective of Step 2). The radiance of the reflected light in units of photons $\cdot \text{m}^{-2} \cdot \text{sr}^{-1} \cdot \text{s}^{-1}$ is given by

$$I = r \left[\left(\int_{\lambda_{min}}^{\lambda_{max}} \frac{\lambda}{hc} S_{ref,\lambda} d\lambda \right) \left(\frac{r_{ref}}{\|\mathbf{r}_{sc}\|} \right)^2 \right] = r S_{ref} \left(\frac{r_{ref}}{\|\mathbf{r}_{sc}\|} \right)^2 \quad (3.110)$$

where r is computed from Eq. 3.92 and S_{ref} is computed from Eq. 3.109.

3.9.2.3 Step 3: Find Number of Photoelectrons Generated at Each Illuminated Pixel

The third and final step is to determine the number of photoelectrons generated at each pixel location. To do this requires the conversion of radiance at the source (surface patch on planet or moon) to irradiance at the observer

(each individual pixel location). The irradiance at each pixel location may be used to compute the number photoelectrons generated at that pixel, N_{pe} , from Eq. 3.26.

Begin by taking a closer look at the relation between radiance, I , and irradiance, S . Suppose light from an infinitesimal patch on a source, dA_s , is absorbed by a second infinitesimal patch on the observer, dA_o . If the geometry is as shown in Fig. 3.18, then let s be the distance between the source and the observer, and $d\Omega$ be the solid angle subtended by the observing patch dA_o from the viewpoint of the source. Under these conditions, the definition of radiance and irradiance give

$$S = I d\Omega \quad (3.111)$$

To convert the irradiance, S , to power, P , the area of the projection of the patch dA_s onto a plane perpendicular to the line-of-sight vector is required. Therefore, define θ_s as the angle between the line-of-sight vector and the source

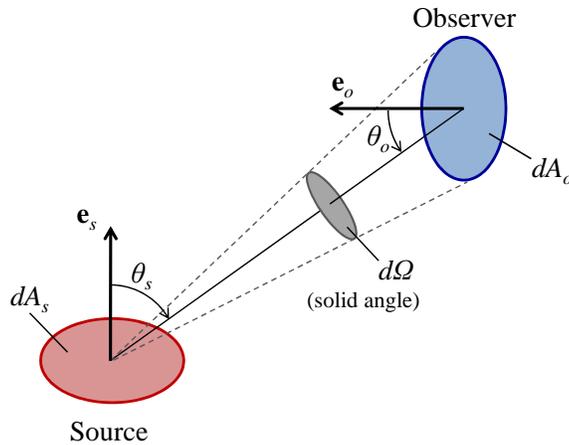


Figure 3.18: Geometry between source and observer.

patch surface normal and define θ_o as the angle between the line-of-sight vector and the observer patch surface normal. If the patch dA_s is small then the projected area used to convert S to P is approximately $dA_s \cos\theta_s$. Under these conditions, geometry will show that

$$P = I dA_s \cos\theta_s d\Omega \quad (3.112)$$

Here, P represents the power emitted by the source over the patch dA_s that will be incident upon the patch dA_o .

Now, recall that a small solid angle, $d\omega$, that subtends the small patch dA at a distance of r from the observer is given by

$$d\omega = \frac{dA \cos\theta}{r^2} \quad (3.113)$$

where θ is the angle between the line-of-sight vector and the patch surface normal.

Therefore, suppose a pixel on a detector observes a patch dA_s on a planet (i.e. dA_s is the projection of the pixel onto the surface of the planet). Additionally assume that light from this source is focused onto this pixel location through a system of lenses, using a camera with an aperture of d_{ap} . The geometry of this situation is as shown in Fig. 3.19 (this figure may be thought of as the appropriate slice of Fig. 3.15). Because the system of lenses focuses the light from the source collected over the entire aperture, the value for $d\Omega$ in Eq. 3.111 and Eq. 3.112 is computed as the solid angle subtended by the camera aperture as seen from the surface of the planet/moon,

$$d\Omega = \frac{dA \cos\theta_o}{\mathbf{s}^T \mathbf{s}} = \frac{\pi}{4} \frac{d_{ap}^2 \cos\theta_o}{\mathbf{s}^T \mathbf{s}} \quad (3.114)$$

The only remaining variable required to solve Eq. 3.112, is an expression for dA_s . Because the solid angle $d\omega$ in Fig. 3.19 must be the same on both sides of the lens, applying Eq. 3.113 shows that

$$d\omega = \frac{dA_o \cos\theta_o}{(Z_l/\cos\theta_o)^2} = \frac{dA_s \cos\theta_s}{\mathbf{s}^T \mathbf{s}} \quad (3.115)$$

Therefore, it is straightforward to show that

$$dA_s = dA_o \frac{\mathbf{s}^T \mathbf{s}}{Z_l^2} \frac{\cos^3\theta_o}{\cos\theta_s} \quad (3.116)$$

Substituting the results of Eq.3.114 and Eq. 3.116 into Eq. 3.112

$$P = I \left(dA_o \frac{\mathbf{s}^T \mathbf{s}}{Z_l^2} \frac{\cos^3\theta_o}{\cos\theta_s} \right) \cos\theta_s \left(\frac{\pi}{4} \frac{d_{ap}^2 \cos\theta_o}{\mathbf{s}^T \mathbf{s}} \right) \quad (3.117)$$

Simplifying this result yields

$$P = I \frac{\pi}{4} dA_o \left(\frac{d_{ap}}{Z_l} \right)^2 \cos^4\theta_o \quad (3.118)$$

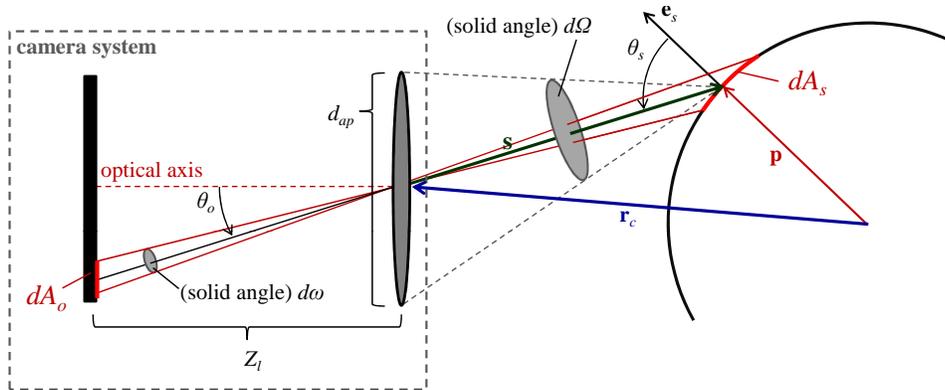


Figure 3.19: Geometry for a camera system viewing a section of a planet. Image not to scale and angles are exaggerated for clarity.

Unfortunately, the system of lenses used to focus the light onto the pixel is not a lossless system (as was assumed in creating Eq. 3.118). Therefore, as was done in the starlight model from Eq. 3.81, introduce T as the transmittance of the optics to account for losses associated with the optical system. Therefore, if the irradiance incident on the pixel is desired, then

$$S = \frac{PT}{dA_o} = IT \frac{\pi}{4} \left(\frac{d_{ap}}{Z_l} \right)^2 \cos^4 \theta_o \quad (3.119)$$

If the value of I is computed from Eq. 3.110, then the units of S are photons \cdot m⁻² \cdot s⁻¹. Therefore, the number of photoelectrons generated by a pixel, N_{pe} , may be computed by Eq. 3.26.

3.10 Intentional Defocusing

3.10.1 Star Centroiding

The simplest star centroiding algorithm consists of taking the weighted average (weighted by pixel intensity) of the u -location and v -location of pixels in the neighborhood of the star. A window is drawn around the sensed star with the upper left pixel of the window being defined by coordinates $[u_{w0}, v_{w0}]$. Remember that the average signal generated by a set of dark pixels should be subtracted from each pixel before performing the centroiding algorithm.

Therefore, let the total intensity in a window of interest about a star be given by

$$I_{w,total} = \sum_{i=u_{w0}}^{u_{w0}+n_u} \sum_{j=v_{w0}}^{v_{w0}+n_v} I_{dig}(u_i, v_j) \quad (3.120)$$

where $I_{dig}(u_i, v_j)$ is the pixel intensity at location $[u_i, v_j]$ in the digital image (as given by the mapping described in Eq. 3.19), n_u is the number of pixels in the window along the u -axis, and n_v is the number of pixels in the window along the v -axis. Then the weighted average is computed by

$$u_c = \frac{1}{I_{w,total}} \sum_{i=u_{w0}}^{u_{w0}+n_u} \sum_{j=v_{w0}}^{v_{w0}+n_v} I_{dig}(u_i, v_j) u_i \quad (3.121)$$

$$v_c = \frac{1}{I_{w,total}} \sum_{i=u_{w0}}^{u_{w0}+n_u} \sum_{j=v_{w0}}^{v_{w0}+n_v} I_{dig}(u_i, v_j) v_j \quad (3.122)$$

3.10.2 Determining Optimal Amount of Defocusing

Stars will normally appear as point sources in a focused image, thus only illuminating one or two pixels. For example, the star R Doradus has the largest apparent diameter in the night sky, with an apparent diameter of 0.057 arcsec.[119] Assuming a camera with a focal length of 50 mm, this would create a spot with a diameter of approximately 0.03 μm diameter in an ideal system. In a real system, but still assuming the optics are good enough to create a diffraction limited system, a spot the size of the Airy disk would be created ($d_{airy} \approx 2.7\mu\text{m}$ for an f-number of 1.5).

With so few pixels illuminated, it is difficult to obtain an accuracy beyond what is defined by the pixel dimensions. The quantization would generate an error in the estimate of the centroid location with a standard deviation of about $1/\sqrt{12}$ pixels in the u -direction and $1/\sqrt{12}$ pixels in the v -direction (using the same notation as in Eq. 3.16). When the image is defocused, multiple pixels are illuminated and centroiding techniques may be used to obtain

subpixel accuracy. Historically, these techniques have been shown to provide one to two orders of magnitude improvement over the accuracy defined by the pixel dimensions. This is shown pictorially in Fig. 3.20.

Determining the optimal blur size requires some care. Here, the optimal blur size is defined as the amount of blur that minimizes the error associated with the star centroid estimate. Because the estimated centroid is dependent on the quality of the image data, the signal-to-noise ratio becomes an important parameter. Let the signal-to-noise ratio be defined as the ratio of the total number of photoelectrons generated by a star (i.e. N_{pe} in Eq. 3.26) to the average noise in a single pixel. This is where the amount of noise in the CCD/CMOS sensor enters into the performance of an optical sensor for navigation applications. To quantify this effect and determine the optimal

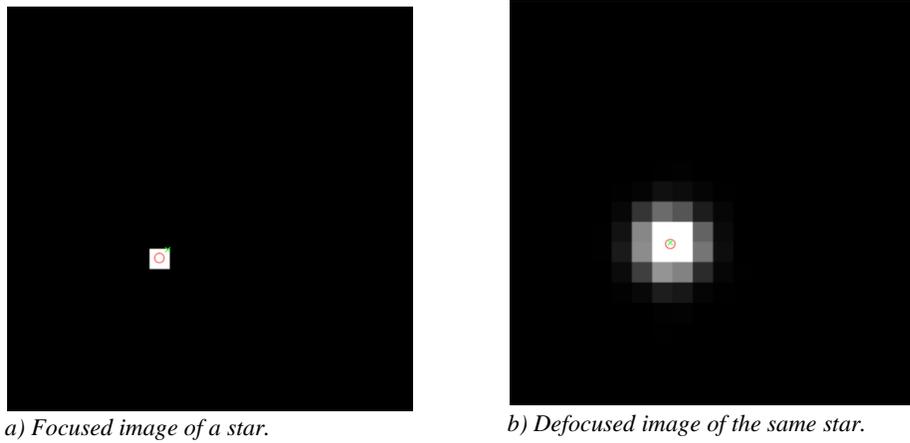


Figure 3.20: Defocusing an image can enable subpixel accuracy through centroiding. Green 'x' shows true star location and red circle shows centroid estimate.

blur size, a Monte Carlo analysis was performed to select the best blur size over a range of signal-to-noise ratios. The size of the blur will be defined by the standard deviation of the bivariate Gaussian PSF described in Eq. 3.59. Additionally, it is assumed that a significant number of pixels illuminated by a given star are not saturated. Performing the Monte Carlo analysis results in RMS errors in the estimated centroid location as shown in Fig. 3.21. From this figure, the optimal blur size is approximately 0.5-1.0 pixels and the resulting RMS centroid can be consistently determined to less than a tenth of a pixel.

Now, consider the situation where pixel saturation becomes an important factor. For ease of discussion, assume a constant sensor noise level and simply consider stars of increasing brightness. As pixels become saturated, information is lost and the error associated with the centroid estimate increases. Such an effect would cause increased error in very strong signals, resulting in increased centroid errors on the right-hand side of Fig. 3.21. Therefore, instead of having the low centroid error region extend infinitely far off the right-hand side of the plot, a valley will form such that stars of intermediate brightness will provide the best performance. Stars that are too dim will have large centroid error due to detector noise, while stars that are too bright will have large centroid errors due to pixel saturation.

3.11 Creating a Simulated Image

Now that that the ideal measurement model and individual noise sources have been presented, a procedure may be developed to simulate the image gen-

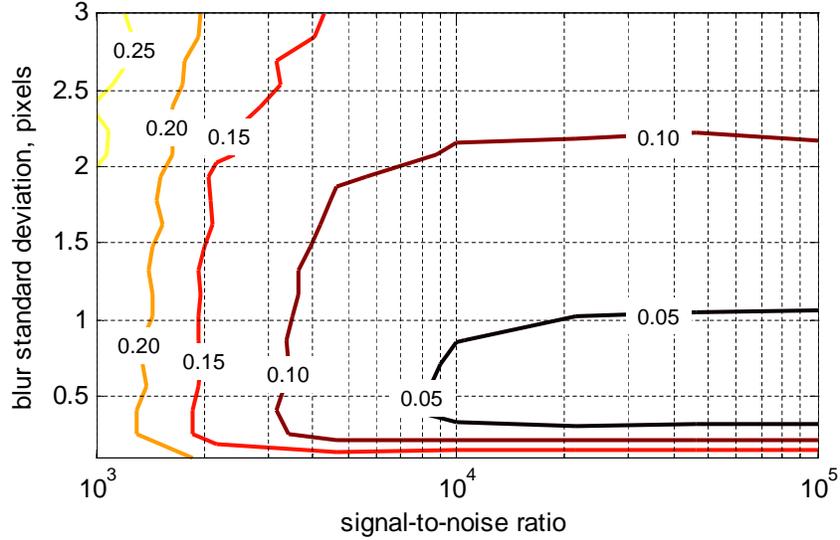


Figure 3.21: RMS centroid error along x-axis (contours in units of pixels) as a function signal-to-noise ratio and blur standard deviation.

erated by an optical sensor.

A typical OPNAV image will contain both a planet (or moon) and stars. Begin by computing the pixels illuminated by the planet or moon. Next compute the location of each star in the camera FOV. Using Eq. 3.89, if the line-of-sight vector to the current star of interest is $(\mathbf{e}_{star})_I$ and the following is found to be true,

$$\left[(\mathbf{e}_{star})_I^T \mathbf{A} \mathbf{r}_c + \mathbf{r}_c^T \mathbf{A} (\mathbf{e}_{star})_I \right]^2 - 4 \left[(\mathbf{e}_{star})_I^T \mathbf{A} (\mathbf{e}_{star})_I \right] \left[\mathbf{r}_c^T \mathbf{A} \mathbf{r}_c - 1 \right] \geq 0 \quad (3.123)$$

then the planet will be occluding the star and the star should not be included in the simulated image. Once the ideal focused image has been created, it may need to be blurred if the camera is not focused (as was discussed above,

the image not being perfectly focused may be either intentional or unintentional). Blur may be introduced by convolving a linear Gaussian filter with the perfectly focused version of the image.

If the digital image is given by $I_{dig}(u, v)$ as described in Eq. 3.19 and the Gaussian filter is given by G , then the blurred image, $H_{dig}(u, v)$, is given by

$$H_{dig}(u, v) = G * I_{dig}(u, v) \quad (3.124)$$

where $*$ denotes the convolution operator.[87] A detailed discussion of image convolution and linear filters is given in Section 4.2.1. Depending on the level of accuracy required in the simulation, the filter may be applied to an image of higher resolution and then downsampled to the desired resolution to better simulate the behavior associated with defocus.

After the image has been blurred, noise may be added to the image. To each pixel, explicitly add random noise associated with shot noise, the noise floor, reset noise, and pattern noise,

$$H_{dig}(u, v) = H_{dig}(u, v) + \sigma_{sim}\epsilon(u, v) \quad (3.125)$$

where $\epsilon(u, v)$ is zero-mean white noise with $E[\epsilon(u, v)^2] = 1$ and σ_{sim} is given by

$$\sigma_{sim} = [\sigma_{shot}^2 + \sigma_{floor}^2 + \sigma_{reset}^2 + \sigma_{pattern}^2]^{\frac{1}{2}} \quad (3.126)$$

Note that unlike σ_{sys} in Eq. 3.60, the ADC noise is not explicitly added to the signal. Instead, as a final step, the value at each pixel is quantized as described in Eq. 3.70. This will result in the same amount of noise, but explicitly

performing the quantization provides a better simulation of the discrete nature of the sensor's digital output.

3.12 Measurement Covariance Matrix for Optical Line-of-Sight Observations

3.12.1 Classic Approach for Line-of-Sight Unit Vector Covariance

Suppose the true line-of-sight unit vector, expressed in frame I , to the i -th observed object is given by $(\mathbf{e}_i)_I$. Now suppose a measured line-of-sight unit vector to this same object is available in the spacecraft camera frame, $(\tilde{\mathbf{e}}_i)_C$. If the orientation of the camera with respect to the body frame is known, then the measured line-of-sight unit vector in the body frame is given by

$$(\tilde{\mathbf{e}}_i)_B = \mathbf{T}_B^C (\tilde{\mathbf{e}}_i)_C \quad (3.127)$$

Because the line-of-sight is constrained to be a unit vector, error in its direction should formally be represented as a rotation. It is, however, convenient to represent this as an additive error in many cases,

$$(\tilde{\mathbf{e}}_i)_C = \mathbf{T}_{\tilde{C},i}^C (\mathbf{e}_i)_C = (\mathbf{e}_i)_C + \boldsymbol{\epsilon}_i \quad (3.128)$$

where $\mathbf{T}_{\tilde{C},i}^C$ is the rotation matrix that rotates the true line-of-sight in the camera frame to the measured line-of-sight in the camera frame¹⁰ and $\boldsymbol{\epsilon}_i$ is the measurement error in the observed unit vector. For $(\tilde{\mathbf{e}}_i)_C$ to remain a unit

¹⁰This means that $(\tilde{\mathbf{e}}_i)_C = (\mathbf{e}_i)_{\tilde{C},i}$. In other words, the \tilde{C} frame is a frame close to the true camera frame that produces the observed line-of-sight measurement.

vector to first order, the measurement error must be perpendicular to the true observation unit vector:

$$(\mathbf{e}_i)_C^T \boldsymbol{\epsilon}_i \approx 0 \quad (3.129)$$

If the measurement is further assumed to be unbiased, then

$$E[\boldsymbol{\epsilon}_i] = 0 \quad (3.130)$$

Now, if the angle between the C frame and the \tilde{C} frame is small, then $\mathbf{T}_{\tilde{C},i}^C$ may be approximated as described in Eq. 3.5. Substituting this result yields:

$$(\tilde{\mathbf{e}}_i)_C = [\mathbf{I}_{3 \times 3} - [\boldsymbol{\delta\phi} \times]] (\mathbf{e}_i)_C \quad (3.131)$$

where $\boldsymbol{\delta\phi}$ is a small angle that rotates the true unit vector direction to the measured unit vector direction. Now, substituting Eq. 3.131 into Eq. 3.128 and solving for $\boldsymbol{\epsilon}_i$,

$$\boldsymbol{\epsilon}_i = -[\boldsymbol{\delta\phi} \times] (\mathbf{e}_i)_C = [(\mathbf{e}_i)_C \times] \boldsymbol{\delta\phi} \quad (3.132)$$

Therefore, the error covariance is given by

$$E[\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T] = E\left[[(\mathbf{e}_i)_C \times] \boldsymbol{\delta\phi} \boldsymbol{\delta\phi}^T [(\mathbf{e}_i)_C \times]^T\right] \quad (3.133)$$

$$E[\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T] = [(\mathbf{e}_i)_C \times] E[\boldsymbol{\delta\phi} \boldsymbol{\delta\phi}^T] [(\mathbf{e}_i)_C \times]^T \quad (3.134)$$

Assuming the covariance of the small angle rotation is given by

$$E[\boldsymbol{\delta\phi} \boldsymbol{\delta\phi}^T] = \sigma_{\phi,i}^2 \mathbf{I}_{3 \times 3} \quad (3.135)$$

where $\sigma_{\phi,i}^2$ is the angular variance (in radians) of the measurement error, the measurement covariance is may be rewritten as

$$E [\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T] = \sigma_{\phi,i}^2 [(\mathbf{e}_i)_C \times] [(\mathbf{e}_i)_C \times]^T \quad (3.136)$$

$$E [\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T] = \sigma_{\phi,i}^2 [(\mathbf{e}_i)_C \times] [- (\mathbf{e}_i)_C \times] \quad (3.137)$$

Recalling a useful property of the cross product matrix,

$$[\boldsymbol{\alpha} \times] [\boldsymbol{\beta} \times] = \beta \boldsymbol{\alpha}^T - \boldsymbol{\beta}^T \boldsymbol{\alpha} \mathbf{I}_{3 \times 3} \quad (3.138)$$

the measurement covariance in the camera frame may be rewritten as

$$\mathbf{R}_{C,i} = E [\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T] = \sigma_{\phi,i}^2 [\mathbf{I}_{3 \times 3} - (\mathbf{e}_i)_C (\mathbf{e}_i)_C^T] \quad (3.139)$$

It is straightforward to rotate this covariance matrix into the body frame,

$$\mathbf{R}_{B,i} = \mathbf{T}_B^C \mathbf{R}_{C,i} \mathbf{T}_C^B \quad (3.140)$$

Distributing the \mathbf{T}_B^C yields the measurement covariance that is typically seen in the literature[120] (although this derivation arrives at this result from a camera-centric perspective instead of assuming that the measurement is already in the body frame),

$$\mathbf{R}_{B,i} = \sigma_{\phi,i}^2 [\mathbf{I}_{3 \times 3} - (\mathbf{e}_i)_B (\mathbf{e}_i)_B^T] \quad (3.141)$$

This simple model, which is widely used in the literature, provides no means of altering the shape of the covariance matrix beyond the line-of-sight direction. It is also unclear how to best pick a value for $\sigma_{\phi,i}$. As was discussed in earlier

sections, errors in the sensor and in construction of the camera will lead to behavior that is difficult to model with Eq. 3.139.

Unfortunately, the matrix \mathbf{R} in Eq. 3.139 is singular (it is a 3×3 matrix with a rank of two), and an actual inverse may not be computed. As will become clear later, the inverse measurement covariance matrix, \mathbf{R}^{-1} , for line-of-sight unit vector observations is needed for some of the attitude filtering techniques. Therefore, computing the Moore-Penrose pseudoinverse¹¹ of \mathbf{R} yields

$$\text{“}\mathbf{R}^{-1}\text{”} = \mathbf{R}^\dagger = \frac{1}{\sigma_{\phi,i}^2} \left[\mathbf{I}_{3 \times 3} - (\mathbf{e}_i)_C (\mathbf{e}_i)_C^T \right] \quad (3.142)$$

where the \dagger symbol indicates the pseudoinverse. The use of the pseudoinverse in this application has been investigated in more detail by Shuster.[121]

3.12.2 Line-of-Sight Unit Vector Covariance from Camera Parameters

The discussion that follows presents new results that directly relate errors in measurements made by the camera to error in the i -th line-of-sight unit vector. The method presented here differs from the traditional approach by looking at errors in the line-of-sight unit vectors as a function of camera error sources. This provides a better understanding of the measurement error characteristics that are critical in improving filter performance.

¹¹The easiest way is through Singular Value Decomposition (SVD). Let the SVD of \mathbf{R} be given by $\mathbf{R} = \mathbf{U}\mathbf{F}\mathbf{V}^*$, where \mathbf{F} is a diagonal matrix. In this case, the Moore-Penrose pseudoinverse is given by $\mathbf{R}^\dagger = \mathbf{U}\mathbf{F}^\dagger\mathbf{V}^*$, where \mathbf{F}^\dagger is a diagonal matrix containing the reciprocal values of the non-zero entries in \mathbf{F} .

Begin by adding noise to the camera measurement model given in Eq. 3.22,

$$(\tilde{\mathbf{e}}_i)_C = \left(\tilde{X}_i^2 + \tilde{Y}_i^2 + \hat{Z}_l^2 \right)^{-\frac{1}{2}} \begin{bmatrix} -\tilde{X}_i \\ -\tilde{Y}_i \\ \hat{Z}_l \end{bmatrix} \quad (3.143)$$

where \tilde{X}_i is the measured value of X_i , \tilde{Y}_i is the measured value of Y_i , and \hat{Z}_l is the estimate of Z_l :

$$X_i = \tilde{X}_i + \delta X_i \quad Y_i = \tilde{Y}_i + \delta Y_i \quad Z_l = \hat{Z}_l + \delta Z_l \quad (3.144)$$

such that δX_i is the error in the measured value of X_i , δY_i is the error in the measured value of Y_i , δZ_l is the error in the estimate of Z_l . Making the appropriate substitutions in Eq. 3.143,

$$(\mathbf{e}_i)_{\hat{C}} = \left[(X_i - \delta X_i)^2 + (Y_i - \delta Y_i)^2 + (Z_l - \delta Z_l)^2 \right]^{-\frac{1}{2}} \begin{bmatrix} -X_i + \delta X_i \\ -Y_i + \delta Y_i \\ Z_l - \delta Z_l \end{bmatrix} \quad (3.145)$$

Expanding the denominator in Eq. 3.145 and retaining only first-order terms,

$$(\tilde{\mathbf{e}}_i)_C \approx \left(X_i^2 - 2X_i\delta X_i + Y_i^2 - 2Y_i\delta Y_i + Z_l^2 - 2Z_l\delta Z_l \right)^{-\frac{1}{2}} \begin{bmatrix} -X_i + \delta X_i \\ -Y_i + \delta Y_i \\ Z_l - \delta Z_l \end{bmatrix} \quad (3.146)$$

Proceed by expanding the denominator as a Taylor series and, as before, retain only first-order terms

$$(\tilde{\mathbf{e}}_i)_C \approx (\mathbf{e}_i)_C + (X_i^2 + Y_i^2 + Z_l^2)^{-\frac{3}{2}} \left\{ \begin{bmatrix} -X_i \\ -Y_i \\ Z_l \end{bmatrix} \times \left(\begin{bmatrix} \delta X_i \\ \delta Y_i \\ -\delta Z_l \end{bmatrix} \times \begin{bmatrix} -X_i \\ -Y_i \\ Z_l \end{bmatrix} \right) \right\} \quad (3.147)$$

To compact notation, let $D_i = (X_i^2 + Y_i^2 + Z_i^2)^{\frac{1}{2}}$ and $\boldsymbol{\delta\gamma}_i = [\delta X_i \delta Y_i -\delta Z_i]^T$.

The above expression may now be reduced to

$$(\tilde{\mathbf{e}}_i)_C \approx (\mathbf{e}_i)_C + D_i^{-1} \{(\mathbf{e}_i)_C \times (\boldsymbol{\delta\gamma}_i \times (\mathbf{e}_i)_C)\} \quad (3.148)$$

$$(\tilde{\mathbf{e}}_i)_C \approx (\mathbf{e}_i)_C - D_i^{-1} [(\mathbf{e}_i)_C \times] [(\mathbf{e}_i)_C \times] \boldsymbol{\delta\gamma}_i \quad (3.149)$$

Recalling Eq. 3.128, it is now straightforward to show that

$$\boldsymbol{\epsilon}_i \approx -D_i^{-1} [(\mathbf{e}_i)_C \times] [(\mathbf{e}_i)_C \times] \boldsymbol{\delta\gamma}_i \quad (3.150)$$

Recalling the identity from Eq. 3.138, the above equation may be rewritten as

$$\boldsymbol{\epsilon}_i \approx -D_i^{-1} \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \boldsymbol{\delta\gamma}_i \quad (3.151)$$

From here, proceed straight to the definition of the measurement covariance:

$$\begin{aligned} \mathbf{R}_{C,i} = E [\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T] &\approx E \left[D_i^{-2} \left((\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right) \right. \\ &\quad \left. \boldsymbol{\delta\gamma}_i \boldsymbol{\delta\gamma}_i^T \left((\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right) \right] \end{aligned} \quad (3.152)$$

Recognizing that the only random variable in this equation is $\boldsymbol{\delta\gamma}_i$,

$$\mathbf{R}_{C,i} \approx D_i^{-2} \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] E [\boldsymbol{\delta\gamma}_i \boldsymbol{\delta\gamma}_i^T] \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \quad (3.153)$$

A suitable expression for $E [\boldsymbol{\delta\gamma}_i \boldsymbol{\delta\gamma}_i^T]$ in terms of useful parameters must still be developed. Begin by noting the simple transformation,

$$\boldsymbol{\delta\gamma}_i = \mathbf{G} \begin{bmatrix} \delta X_i \\ \delta Y_i \\ \delta Z_i \end{bmatrix} \quad (3.154)$$

where the transformation matrix \mathbf{G} is

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.155)$$

Now, combining the results of Eq. 3.17 and Eq. 3.49, it is clear that

$$\delta\gamma_i = \mathbf{G}\mathbf{\Pi}_{det} \begin{bmatrix} \frac{1}{s_x} & \frac{\cos(\theta)}{s_x s_y} & 0 \\ 0 & -\frac{\sin(\theta)}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta u_i \\ \delta v_i \\ \delta Z_l \end{bmatrix} \quad (3.156)$$

Again, compact notation by defining the matrix $\mathbf{\Gamma}$

$$\mathbf{\Gamma} = \begin{bmatrix} \frac{1}{s_x} & \frac{\cos(\theta)}{s_x s_y} & 0 \\ 0 & -\frac{\sin(\theta)}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.157)$$

Therefore, the measurement covariance may be rewritten in terms of the statistics of the spot size and focal length fluctuation. If $\delta\boldsymbol{\eta}_i = [\delta u_i \ \delta v_i \ \delta Z_l]^T$ then the unit vector covariance matrix is given by

$$\mathbf{R}_{C,i} \approx D_i^{-2} \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \mathbf{G}\mathbf{\Pi}_{det}\mathbf{\Gamma}E \left[\delta\boldsymbol{\eta}_i \delta\boldsymbol{\eta}_i^T \right] \quad (3.158)$$

$$\mathbf{\Gamma}^T \mathbf{\Pi}_{det}^T \mathbf{G}^T \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right]$$

This result given in Eq. 3.158 is the general expression for the measurement covariance expressed in the camera frame as a function of measurable errors in the camera system. One of the great benefits of arriving at this expression is a deeper understanding of the physical implications and sensor assumptions required to arrive at the traditional measurement covariance given in Eq. 3.139 (or Eq. 3.141).

3.12.3 Comparison of New Covariance Model with Results from Literature

What follows is a brief comparison of the new covariance model with other results from the literature. The first result that merits a comparison is the traditional unit vector covariance model (sometimes called the QUEST measurement model) given in Eq. 3.139. To arrive at the traditional result from the new covariance model of Eq. 3.158, one must first assume that the detector plane is perfectly perpendicular to the true camera optical axis (i.e. $\mathbf{\Pi}_{det} = \mathbf{I}_{3 \times 3}$) and that array axes are perfectly orthogonal,

$$\mathbf{\Gamma} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & -\frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.159)$$

Additionally if one assumes that all the errors are physically of the same size, i.e. $\frac{\sigma_u}{s_x} = \frac{\sigma_v}{s_y} = \sigma_{Zl}$, then it can be shown that

$$E [\delta\gamma_i \delta\gamma_i^T] = \mathbf{G} \mathbf{\Pi}_{det} \mathbf{\Gamma} E [\delta\eta_i \delta\eta_i^T] \mathbf{\Gamma}^T \mathbf{\Pi}_{det}^T \mathbf{G}^T = \sigma_{\gamma,i}^2 \mathbf{I}_{3 \times 3} \quad (3.160)$$

This transforms Eq. 3.158 back to the form of Eq. 3.153. Under this condition, the traditional result may be exactly recovered. Therefore, proceeding with the assumption of Eq. 3.160:

$$\mathbf{R}_{C,i} \approx \sigma_{\gamma,i}^2 D_i^{-2} \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \quad (3.161)$$

Simplifying the above expression yields

$$\mathbf{R}_{C,i} \approx \sigma_{\gamma,i}^2 D_i^{-2} \left[\mathbf{I}_{3 \times 3} - (\mathbf{e}_i)_C (\mathbf{e}_i)_C^T \right] \quad (3.162)$$

Under a small angle approximation where $\tan\phi \approx \sin\phi \approx \phi$, it is clear that

$$\sigma_{\phi,i}^2 \approx \frac{\sigma_{\gamma,i}^2}{D_i^2} \quad (3.163)$$

Finally, the traditional result shown in Eq. 3.139 becomes apparent:

$$\mathbf{R}_{C,i} \approx \sigma_{\phi,i}^2 \left[\mathbf{I}_{3 \times 3} - (\mathbf{e}_i)_C (\mathbf{e}_i)_C^T \right] \quad (3.164)$$

Before proceeding, note that the simplification of Eq. 3.160 makes two important assumptions: (1) camera measurement errors in the u -direction and v -direction are uncorrelated and (2) that the scale of the error in estimating the centroid on the i -th star on the camera is the same as the error in the estimate of separation between the lens and the detector. Depending on the camera used, this second assumption may not be very good. Although there may be a bias in the estimate of Z_i (which is modeled in $\mathbf{\Pi}_{det}$), the value of Z_i should not be changing from measurement to measurement (i.e. one would expect $\sigma_{Z_i} \approx 0$).

Therefore, an alternative case of interest is the one where $\sigma_{Z_i} \approx 0$.

Under this scenario,

$$E [\boldsymbol{\delta\eta}_i \boldsymbol{\delta\eta}_i^T] = \begin{bmatrix} \sigma_u^2 & \rho_{uv}\sigma_u\sigma_v & 0 \\ \rho_{uv}\sigma_u\sigma_v & \sigma_v^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.165)$$

is chosen instead of the identity shown in Eq. 3.160. The variable ρ_{uv} is the correlation coefficient between δu and δv . To simplify, let \mathbf{R}_{uv} be given by

$$\mathbf{R}_{uv} = E \left[\begin{bmatrix} (u - \hat{u}) \\ (v - \hat{v}) \end{bmatrix} \begin{bmatrix} (u - \hat{u}) \\ (v - \hat{v}) \end{bmatrix}^T \right] \quad (3.166)$$

If errors in u and v are uncorrelated, then $E[(u - \hat{u})(v - \hat{v})] = 0$, and \mathbf{R}_{uv} is a diagonal matrix. The u - v covariance matrix describes the shape of the error ellipse of the centroid on the observed object on the detector plane. Therefore, Eq. 3.165 may be rewritten as

$$E[\delta\boldsymbol{\eta}_i \delta\boldsymbol{\eta}_i^T] = \begin{bmatrix} \mathbf{I}_{2 \times 2} \\ \mathbf{0}_{1 \times 2} \end{bmatrix} \mathbf{R}_{uv} \begin{bmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 1} \end{bmatrix} \quad (3.167)$$

Therefore, one should expect that

$$\begin{aligned} \mathbf{R}_{C,i} \approx \frac{1}{D_i^2} & \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \\ & \boldsymbol{\Pi}_{det} \boldsymbol{\Gamma} \begin{bmatrix} \mathbf{I}_{2 \times 2} \\ \mathbf{0}_{1 \times 2} \end{bmatrix} \mathbf{R}_{uv} \begin{bmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 1} \end{bmatrix} \boldsymbol{\Gamma}^T \boldsymbol{\Pi}_{det}^T \\ & \left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \end{aligned} \quad (3.168)$$

Note that the structure of this problem allows for the -1 in the third element of the \mathbf{G} matrices to be multiplied by only zeros. Because the upper 2×2 portion of this matrix is the identity matrix, \mathbf{G} may be eliminated from the equation.

For cameras with a relatively narrow FOV, note that $(\mathbf{e}_i)_C \approx [0 \ 0 \ 1]^T$.

Therefore the following identity is approximately true,

$$\left[(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3} \right] \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.169)$$

Therefore, the elements in the covariance matrix associated with σ_{Z_i} would be nearly eliminated even if they were nonzero for cameras with a narrow FOV. The traditional covariance in Eq. 3.139 (or Eq. 3.141) may now be derived

for this case following the same approach as for the first case. Note, however, that the approximation of Eq. 3.169 makes the relationship derived from Eq. 3.165 significantly weaker than the relationship derived from Eq. 3.160.

The case where $\sigma_{z_i} \approx 0$ is also relevant with respect to the covariance model introduced by Cheng et al. in [122]. In their model, they show that

$$\mathbf{R}_{C,i} = \mathbf{J}_i \mathbf{R}_{xy,i} \mathbf{J}_i^T \quad (3.170)$$

where \mathbf{R}_{xy} is the covariance of the point on the detector plane (therefore, this is simply \mathbf{R}_{uv} mapped from the D frame into the C frame) and \mathbf{J}_i is given by

$$\mathbf{J}_i = \frac{1}{D} \begin{bmatrix} -\mathbf{I}_{2 \times 2} \\ \mathbf{0}_{1 \times 2} \end{bmatrix} - \frac{1}{D^2} (\mathbf{e}_i)_C [x_i \quad y_i] \quad (3.171)$$

To provide a comparison with the new covariance model, it is straightforward to show that

$$\begin{aligned} \mathbf{J}_i &= \frac{1}{D} \begin{bmatrix} -\mathbf{I}_{2 \times 2} \\ \mathbf{0}_{1 \times 2} \end{bmatrix} - \frac{1}{D^2} (\mathbf{e}_i)_C [x_i \quad y_i] \\ &= \frac{1}{D} [(\mathbf{e}_i)_C (\mathbf{e}_i)_C^T - \mathbf{I}_{3 \times 3}] \begin{bmatrix} \mathbf{I}_{2 \times 2} \\ \mathbf{0}_{1 \times 2} \end{bmatrix} \end{aligned} \quad (3.172)$$

This means that the new covariance model given in Eq. 3.168 collapses to the model from Eq. 3.170 if the errors are expressed in the C frame (instead of the D frame) and if the other camera error sources are ignored (e.g. array axes not perfectly perpendicular to each other, detector array not perfectly perpendicular to camera boresight, etc.).

Additionally, in this context it is worth pointing out that [122] and

[123] prove that Eq. 3.170 collapses exactly to the form of Eq. 3.139 if¹²

$$\mathbf{R}_{xy,i} = \sigma_\phi^2 \frac{D^2}{Z_i^2} \begin{bmatrix} Z_i^2 + x_i^2 & x_i y_i \\ x_i y_i & Z_i^2 + y_i^2 \end{bmatrix} \quad (3.173)$$

Unlike the observation of Eq. 3.169, this is true even when the observed point is not near the boresight (i.e. it is still valid for wide FOV sensors). Because of the observation in Eq. 3.172, this is also approximately true for the new covariance model introduced in the previous section.

This leads to the observation that for the same size/shape of the error ellipse on the detector plane, the resulting error covariance in the line-of-sight unit vector observation changes as the ellipse is moved around the detector plane. Or, from another perspective, a cone representing the same amount of error in the line-of-sight unit vector direction will project to ellipses of varying sizes as the observed point moves around the detector plane. This is shown graphically in Fig. 3.22.

3.13 Optical Sensor Model Case Study: Defocused Star Tracker

A star tracker is an optical device that projects the image of a star field onto the detector plane and uses the resulting image data to estimate attitude. These devices are commonly seen on spacecraft of all types. Therefore, the

¹²Note that the results given in Eq. 3.171 and Eq. 3.173 are not exactly in the same form as presented in [122] and [123]. The authors of these two references choose to normalize the variables x_i and y_i by the focal length, Z_i . The form shown here does not perform that normalization.

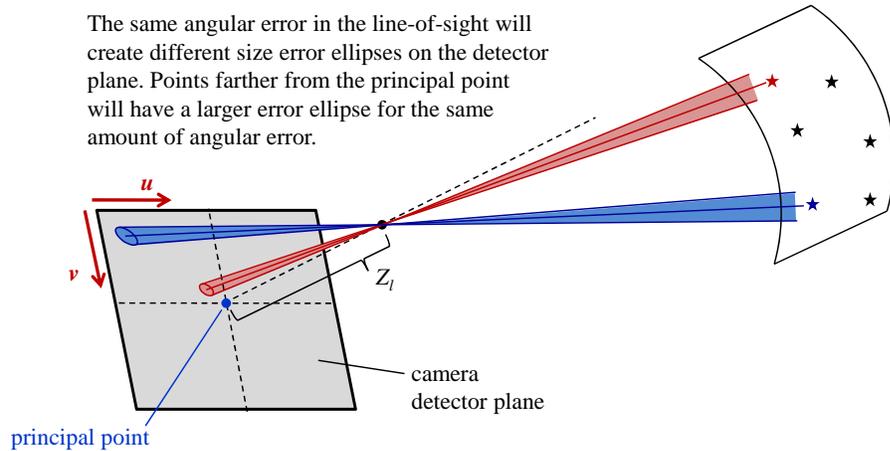


Figure 3.22: Graphical depiction of changing error ellipses on the detector plane for two different line-of-sight unit vector observations with the same amount of angular error.

first case study used to evaluate the optical sensor model is a defocused image of a star field. A detailed discussion of star trackers is provided in Chapter 5.

Suppose a camera with 22 degree FOV, a 6 cm aperture, and a STAR1000 CMOS detector (see Table 3.3 for detector specifications) is to be used for optical attitude determination. To provide a recognizable example, suppose the star tracker captures an image of the constellation Lyra with a 1/30 second exposure. This creates a raw image with a simulated signal as shown in Fig. 3.23. If a threshold is applied such that pixels with an intensity value above τ are set equal to 1 (white) and pixels with an intensity value below τ are set equal to 0 (black), then candidate stars may be easily identified as seen in Fig. 3.24. The use of thresholding is discussed in detail in section 4.2.2. The major stars in Lyra have been labeled in yellow to help the reader identify the

Table 3.3: Summary of performance specifications for Cypress Semiconductor STAR1000 radiation hard CMOS image sensor.[124]

Parameter	Description
Sensitive area format	1024 × 1024 pixels
Pixel size	15 μ m × 15 μ m
Spectral range	400-1000 nm
Average QE × fill factor	20%
Pixel full well capacity	135,000 e ⁻
Dark current density at 293K	223 pA/cm ²
Reset noise	47 e ⁻
Dynamic range	69 dB
Fixed pattern noise	$U_{FPN} \leq 0.56\%$
Photo-response non-uniformity	$U_{PRNU} \leq 3.93\%$
Number of bits	10

constellation. Note that any centroiding algorithm should be applied to the original image (and not to the binary thresholded image).

3.14 Optical Sensor Model Case Study: The MESSENGER Flybys

As a second case study, consider an OPNAV image that contains a planet. To provide a realistic comparison, images were obtained from the MESSENGER Narrow Angle Camera (NAC) during the spacecraft’s 2007 Venus fly-by and 2008 Mercury fly-by. A few of the key specifications for the MESSENGER NAC are provided in Table 3.4. This case study also presents the opportunity to highlight that despite the interesting optical layout for this optical device (see Fig. 3.25), the geometric camera models developed in this section are still valid.

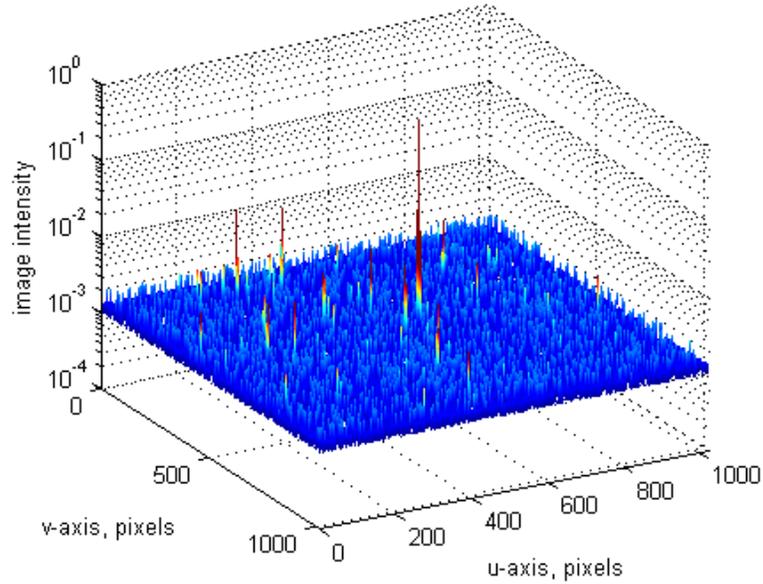


Figure 3.23: Simulated raw image data for star tracker image containing constellation Lyra.

In these results, the simulated images of the planets were generated using the methodology described in Section 3.9.2. This model assumes no surface map. Consequently, no surface features appear in the simulated image - a fact that is more evident in the OPNAV image of Mercury than in the OPNAV image of Venus. Additionally, note that although the detector is a 1024×1024 array, the image from the Mercury fly-by was binned into a 512×512 image. A comparison between a real OPNAV image and a simulated OPNAV image for the 7 June 2007 fly-by of Venus may be seen in Fig. 3.26. A comparison between a real OPNAV image and a simulated OPNAV image for the 15 January 2008 fly-by of Mercury may be seen in Fig. 3.27.

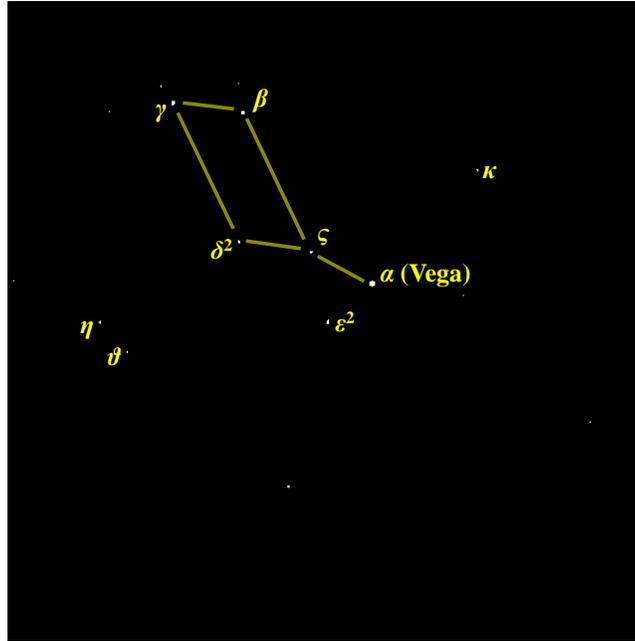


Figure 3.24: Thresholded star tracker image containing constellation Lyra. Major stars in the constellation are labeled for clarity.

These results of Fig. 3.27 and Fig. 3.26 indicate that excellent agreement is observed between the optical sensor models presented in this chapter and reality. This has two important implications. First, images with optical errors and detector noise may be accurately generated. Second, these planet modeling techniques are critical in development the image processing algorithms of Chapter 4 that will extract navigation data from the raw OPNAV images. Specifically, the light intensity models and profiles provide the foundation for generating the filters that will be used to accurately identify the location of a planet/moon horizon.

Table 3.4: Summary of performance specifications for the MESSENGER spacecraft's Narrow Angle Camera.[125]

Parameter	Description
Optics	Off-axis Ritchey-Chrétin
Focal length	550 mm
Aperture	25 mm
F-number	F/22
Field of view	1.5 deg \times 1.5 deg
Exposure time	1 ms to 10 s
Detector	TH7888A CCD
Array size	1024 pixels \times 1024 pixels
Pixel size	14 μ m \times 14 μ m
Number of bits per pixel	12

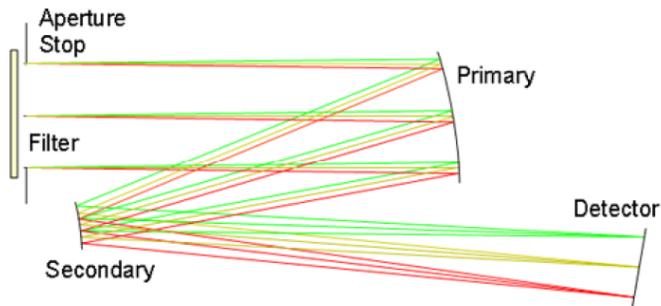


Figure 3.25: Optical layout for the MESSENGER spacecraft's Narrow Angle Camera. With kind permission from Springer Science+Business Media: Fig. 14 from [125].

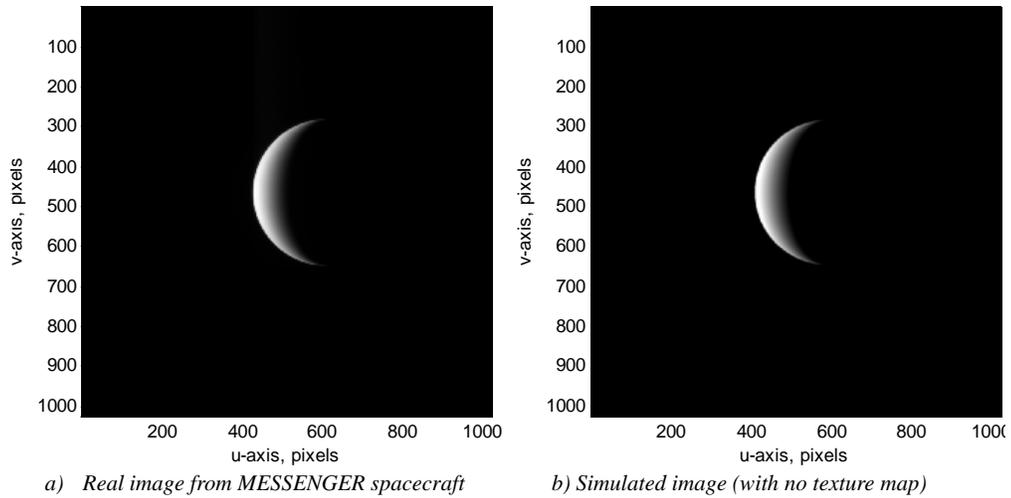


Figure 3.26: Real and simulated OPNAV image from the MESSENGER fly-by of Venus on 7 June 2007. Real image is Product ID EN0089716356M from [93].

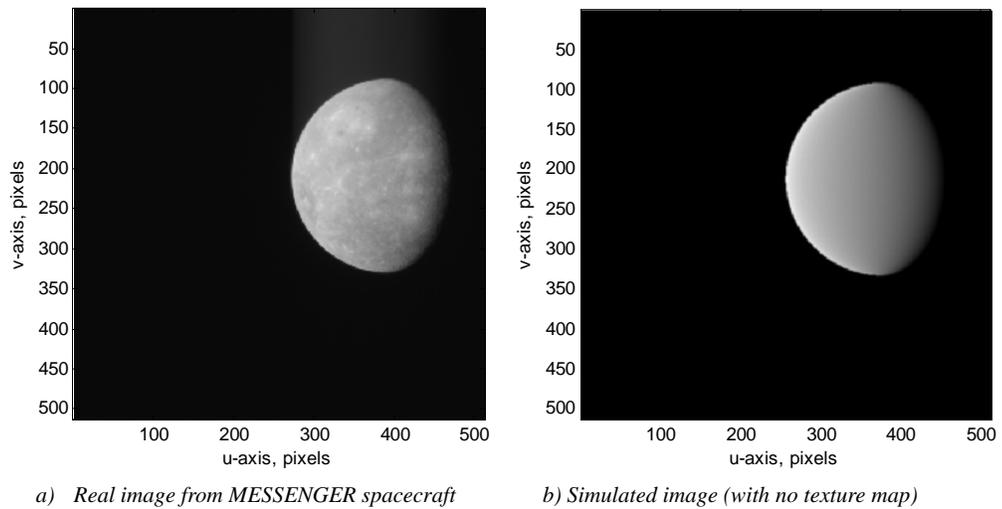


Figure 3.27: Real and simulated OPNAV image from the MESSENGER fly-by of Mercury on 15 January 2008. Real image is Product ID EN0108892844M from [93].

Chapter 4

Image Processing and Generating Optical Navigation Measurements

Image processing algorithms are used to autonomously extract navigation data from a raw image. This chapter begins by introducing the four primary types of optical navigation measurements: (1) apparent diameter and centroid, (2) angle between horizon and reference star, (3) angle between surface feature and reference star, and (4) time of star occultation by planet/moon. Although all four measurement types are introduced, only the first two are actually used here.

The next four sections of this chapter provide the theoretical development for the primary building blocks of the image processing algorithm: Section 4.2 describes basic image processing techniques, Section 4.3 discusses computing the distance to an ellipse, Section 4.4 presents a methodology for fitting of an ellipse to a set of data points, and Section 4.5 describes techniques for robust model fitting in noisy images.

These components are then used to identify candidate stars in Section 4.6 and combined in Section 4.7 to form a new algorithm for the autonomous extraction of navigation observables from a raw OPNAV image containing a

planet or moon. The chapter concludes with a discussion of measurement covariance and the derivation of the measurement sensitivity matrices.

4.1 Summary of Possible Optical Navigation Measurements

This section contains a summary of the four primary types of optical navigation measurements: (1) apparent diameter and centroid, (2) angle between horizon and reference star, (3) angle between surface feature and reference star, and (4) time of star occultation by planet/moon. Although all four measurement types are presented here for completeness, only the apparent diameter and centroid measurement and the star-horizon angle measurement will be used in the subsequent analyses. The reasons for not including the last two measurement types is discussed below.

4.1.1 Apparent Diameter and Centroid

Perhaps the most straightforward optical navigation measurement that can be made is the determination of the planet/moon apparent diameter and centroid. This measurement is unique with respect to the other optical navigation measurements because it is the only one capable of providing a complete position fix.

Suppose that an optical navigation image containing a planet is available. If a sufficiently large portion of the planet is in the image, the direction to the planet center may be found using the image processing techniques dis-

cussed in the chapter. Further, if the true physical size of the planet is known, it may be compared to the observed size of the planet in the image to generate an estimate of the range between the planet and the spacecraft. This procedure is shown notionally in Fig. 4.1. Therefore, the camera position vector with respect to the planet, \mathbf{r}_c , is given by

$$\mathbf{r}_c = -\rho (\mathbf{e}_c)_I \quad (4.1)$$

where ρ is the range from the camera to the planet center and $(\mathbf{e}_c)_I$ is the line-of-sight unit vector from the camera to the planet center expressed in the inertial frame.

Because of the nature of this measurement, different planets and moons will provide different performance. Take the Earth-Moon system as an example. Because the Moon has no atmosphere, the sunlit surface has a crisp horizon, making it relatively easy to accurately detect and measure the horizon of the location. The Earth, on the other hand, has an atmosphere that obscures the horizon to an extent that it may not be seen from orbit. Instead, there is only a fuzzy band with no clear visual feature to support easy mea-

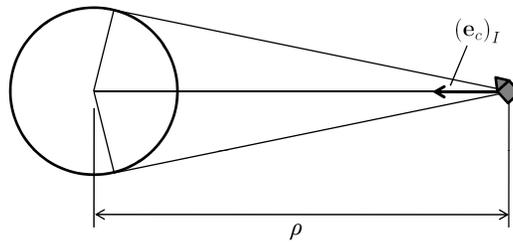


Figure 4.1: Geometry for apparent diameter and centroid measurement.

surement. Despite this difficulty, Apollo astronauts demonstrated their ability to measure a point somewhere in the blurry horizon with repeatability on the order of only a few kilometers.[28] Alternatively, instead of using the visible spectrum to locate the Earth’s horizon, the CO₂ layer of the atmosphere may be observed in the infrared spectrum (14.0-16.3 μm wavelength). The CO₂ layer will form a crisper horizon than achievable in the visible band and is the method of choice for locating the horizon with traditional LEO horizon sensors. A sensor capable of performing this task would be similar to existing infrared static earth sensors (IRSES) that are presently able to achieve apparent diameter estimates with a total RMS error on the order of 0.03 deg (108 arcsec) in LEO.[126]

4.1.2 Angle Between Horizon and Reference Star

The second measurement type used extensively in the following analyses is the angle between the planet horizon and a reference star. Recall from Chapter 2 that star-horizon measurements were used for optical navigation on Gemini, Apollo, and numerous planetary robotic missions.

Suppose an optical navigation image that contains both a reference star and a planet is available. If the line-of-sight unit vector to the star is given by \mathbf{e}_s and the line-of-sight unit vector to the closest point on the planet horizon is given by \mathbf{e}_h , then the angle between these two directions is given by

$$\psi = \text{acos} [\mathbf{e}_s^T \mathbf{e}_h] \quad (4.2)$$

where ψ is the measurement of interest. Unlike the apparent diameter and

centroid measurement, the star-horizon angle measurement cannot create a complete position fix. Because of the geometry of this measurement type (shown in Fig. 4.2), there is a rotational ambiguity about the line-of-sight direction to the reference star. Therefore, a single star-horizon measurement will create a cone of possible spacecraft positions with the axis of symmetry parallel to the line-of-sight direction to the reference star, a cone half-angle equivalent to ψ , and positioned such that the cone surface is tangent to the planet surface. Two star-horizon measurements will create two lines of position. An *a-priori* position estimate or a third star-horizon measurement may be used to determine which line of position the spacecraft is on. It is important to note that no range information may be obtained from star-horizon measurements.

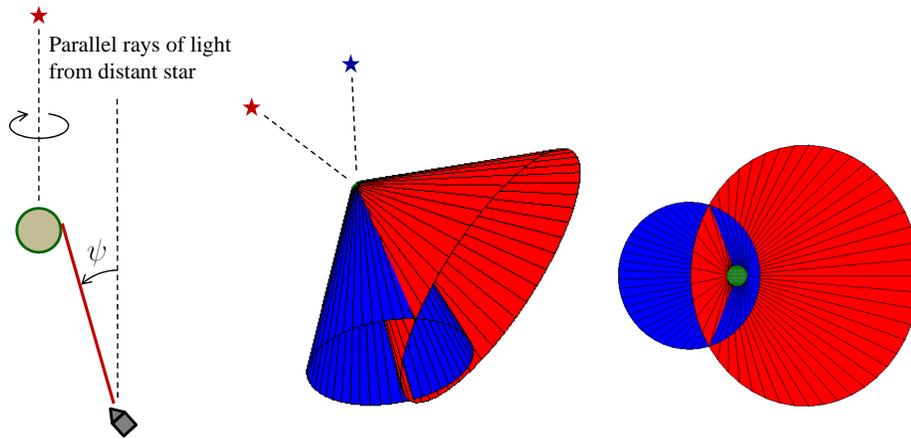


Figure 4.2: Geometry for star-horizon measurement.

4.1.3 Angle Between Surface Feature and Reference Star

The third measurement type is the angle between a surface feature and a reference star. This measurement is very similar to the star-horizon measurement described above. The primary difference is that it requires an extra layer of image processing and pattern recognition to identify specific surface features on a planet or moon.

As before, suppose an optical navigation image that contains both a reference star and a planet is available. If the line-of-sight unit vector to the star is given by \mathbf{e}_s and the line-of-sight unit vector to the surface feature on the planet is given by \mathbf{e}_{SF} , then the angle between these two directions is given by

$$\xi = \text{acos} [\mathbf{e}_s^T \mathbf{e}_{SF}] \quad (4.3)$$

where ξ is the measurement of interest. The geometry of this measurement is shown graphically in Fig. 4.3.

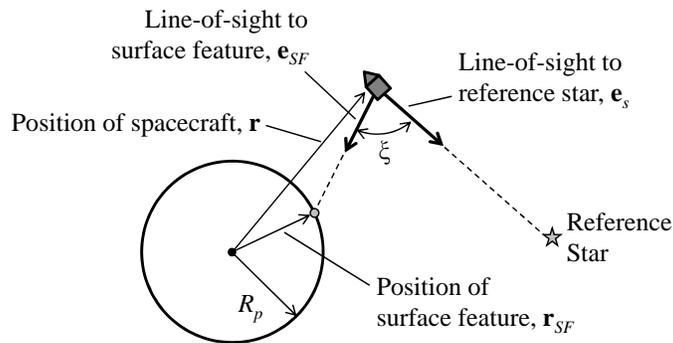


Figure 4.3: Geometry for angular measurement between surface feature and reference star.

If this approach is to be automated, the system must have the capability to identify and track stars and surface features without crew involvement. Automation is clearly necessary if such a system is to be used on a robotic spacecraft. On a crewed spacecraft, automating this process should reduce crew workload, reduce sources of human error, and remove the difficulty the Apollo astronauts experienced with using the space sextant to measure the angle between a surface feature and a reference star. The ability to identify and track stars is a well understood problem and is routinely preformed in star trackers.[19] Planetary surface feature tracking is a significantly more complicated task. A large body of work exists in the literature regarding optical navigation with surface feature tracking and terrain relative navigation for spacecraft in LLO and in lunar landers in the descent/landing mission phase.[127–131]

While this approach may make sense for many celestial bodies that have no atmosphere (e.g. Mercury, Earth’s Moon), it may not work for many planets of interest. Venus, for example, has an atmosphere that completely covers its surface to an extent that no surface features are available from orbit. Alternatively, consider Earth, which has cloud formations that may occlude features of interest or create unpredictable shapes that could confuse an autonomous surface feature recognition algorithm. Similar problems are experienced with the outer planets and some of their moons. This approach was not pursued further in this work for these reasons.

4.1.4 Time of Star Occultation

The fourth and final optical navigation measurement is the occultation of stars by a planet or moon. Here, rather than measuring angles or the location of objects in the image, the time at which stars are eclipsed by the planet/moon is measured. Unfortunately, time is also the independent variable in the dynamic model, which makes directly implementing this measurement into a filter problematic. A number of approaches have been proposed to deal with this problem. Psiaki and Hinks[132] propose a transformation where the apparent altitude of the reference star above the planet surface is measured instead of time (see Fig. 4.4). Occultation occurs when the apparent altitude of the reference star, h_s , is zero. As before, a measurement model may be developed through geometry.

$$h_s = \|\mathbf{r} - (\mathbf{r}^T \mathbf{e}_s) \mathbf{e}_s\| - R_p \quad (4.4)$$

where \mathbf{r} is the position vector from the center of the planet to the spacecraft, \mathbf{e}_s is the unit vector in the direction of the line-of-sight of the reference star, and R_p is the radius of the planet.

In another approach developed by Landgraf et al.,[133] it is assumed that the spacecraft is on a hyperbolic approach trajectory. In their 2006 paper, the authors suggest using two-body orbital mechanics and geometry to determine the true anomaly at which an occultation should occur. Although the following measurement model and derivation differs significantly from that of Landgraf et al., it was inspired by their approach.

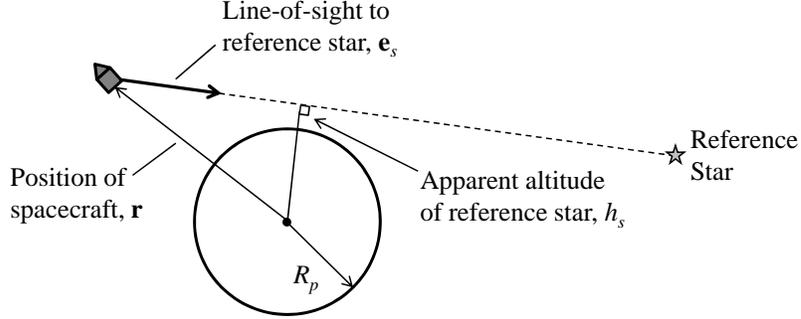


Figure 4.4: Geometry for star occultation.

Consider a star that will be eclipsed by a planet. If the star is sufficiently far away from the planet, the region where the star is eclipsed by the planet forms a cylinder extending away from the reference star (Fig. 4.5). Now suppose the spacecraft of interest is approaching the planet on a hyperbolic orbit. The intersection of this orbit plane and the 3D cylindrical eclipse region will form a 2D elliptical eclipse region in the orbit plane (Fig. 4.6). The intersection of the hyperbolic orbit with the elliptical eclipse region marks the true anomaly where a star occultation begins and ends.

From geometry, it can be shown that the semimajor axis of the eclipse region, $a_{eclipse}$, is given by $a_{eclipse} = R_p/\sin\theta$, where θ is the angle between the vector pointing to the reference star and the spacecraft orbital plane. The semiminor axis, $b_{eclipse}$, is given by $b_{eclipse} = R_p$. A little further geometry demonstrates that:

$$x_{eclipse} = a_{eclipse}\cos(E_{eclipse}) = r_{eclipse}\cos(\nu + \varphi) \quad (4.5)$$

$$y_{eclipse} = a_{eclipse}\sin(E_{eclipse}) = r_{eclipse}\sin(\nu + \varphi) \quad (4.6)$$

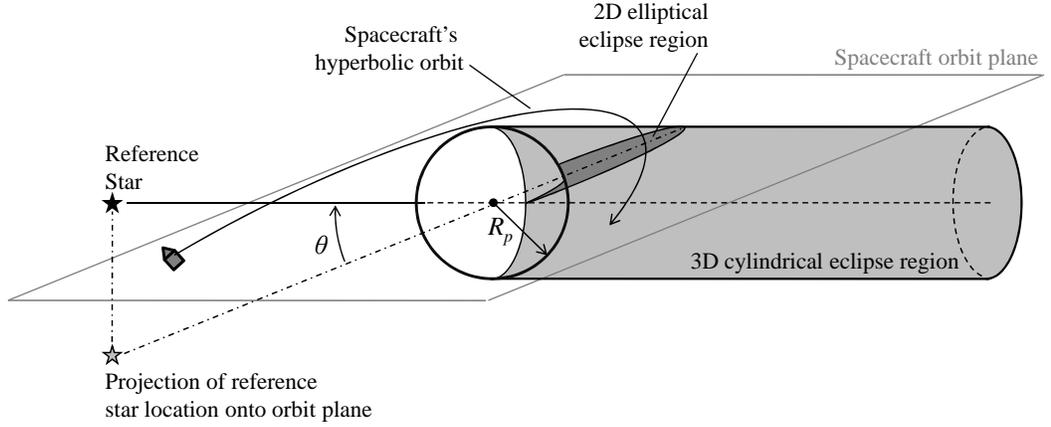


Figure 4.5: Three-dimensional geometry for star occultation.

where $E_{eclipse}$ is the eccentric anomaly of the eclipse ellipse, ν is the true anomaly, and φ is the angle between the orbit periapsis and the projection of the anti-star direction on the orbital plane. Note that all angles in Eq. 4.5 and Eq. 4.6 are taken from the center of the planet, which is located at the geometric center of the eclipse ellipse. Additional manipulation demonstrates that:

$$r_{eclipse}^2 = \frac{R_p^2}{\sin^2\theta + \cos^2\theta \sin^2(\nu + \varphi)} \quad (4.7)$$

where R_p is the radius of the planet (in this case, the Moon). By setting the radius equal to the well known polar equation for a two-body orbit, $r = p/(1 + e\cos(\nu))$, the location at which a specified reference star is eclipsed by the planet is given by the solution to

$$R_p^2(1 + e\cos(\nu))^2 = p^2(\sin^2\theta + \cos^2\theta \sin^2(\nu + \varphi)) \quad (4.8)$$

where p is the orbit semilatus-rectum. Depending on the geometry, zero,

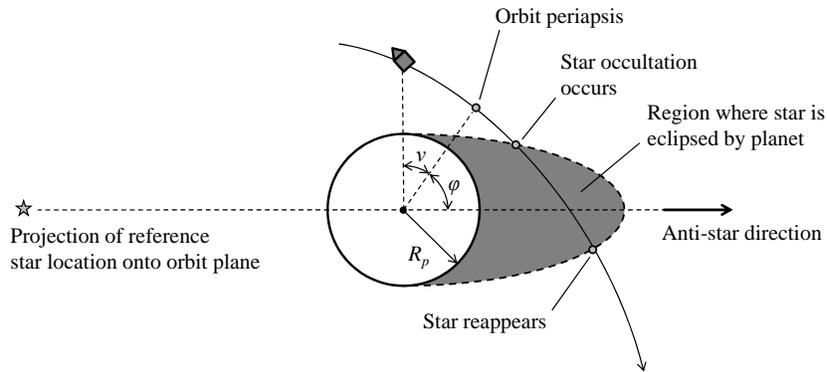


Figure 4.6: Geometry for star occultation as seen in the spacecraft orbit plane.

one, or two real solutions to ν exist. Of particular interest is the case of Eq. 4.8 in which two real solutions to ν exist - one corresponding to the true anomaly where the reference star enters the eclipse of the planet and the other corresponding to the true anomaly where the star leaves the eclipse of the planet.

The time of star occultation measurement is not pursued further in this work. This choice was made for two reasons. First, the direct use of this measurement introduces a stronger dependence on the accuracy of the time estimate and the on-board clock model. Secondly, the formulation of this problem shown in Fig. 4.4 demonstrates that star occultation measurements may be thought of as a star-horizon measurement, with occultation occurring when the star-horizon angle is zero.

4.2 Basic Image Processing

This section describes some basic image processing techniques and introduces the notation to be used in subsequent sections. The techniques described below form the building blocks for the more sophisticated image processing algorithms introduced later in this chapter. Individually, each of these image processing techniques are rather simple, when properly combined, however, they can provide powerful results.

The linear filtering techniques introduced in Section 4.2.1 form the foundation for a template matching based approach for finding candidate stars in an image. The remaining three topics, thresholding (in Section 4.2.2), dilation and erosion (in Section 4.2.3), and connected components analysis (in Section 4.2.4) are used in finding a planet's horizon.

4.2.1 Linear Filters and Templates

As was described in Section 3.2, a digital image may be thought of as a matrix of light intensity values. Let the matrix of values in the the digital image be given by I_{dig} . In some applications it may be desirable to adjust the pixel intensity at a particular location based on the intensity values of neighboring pixels. This process is called filtering.

Suppose that when an image is filtered, the new intensity at each pixel location is given by a weighted sum of the pixels around the pixel of interest. Therefore, define a second matrix, G , that describes the weighting applied to each pixel. The pattern of values in this matrix is frequently called the

“kernel” of the filter. The process of taking this weighted sum is typically called cross-correlation. At any particular pixel location, the cross-correlation may be mathematically described by

$$J_{dig}(u, v) = \sum_{i=-n}^n \sum_{j=-m}^m G(i + n + 1, j + m + 1) I_{dig}(u + i, v + j) \quad (4.9)$$

Applying this relation to all (u, v) pixel locations,

$$J_{dig} = G \star I_{dig} \quad (4.10)$$

where \star denotes the cross-correlation operator, J_{dig} is the filtered image, (u, v) is the image coordinate of the pixel of interest, $2n + 1$ is the number of rows of G , and $2m + 1$ is the number of columns of G . Here, it is interesting to note that when a filter, G , is correlated with an impulse in an image, the resulting filtered image, J_{dig} , contains a flipped (in both u and v) version of the kernel. Therefore, suppose the kernel is flipped before being correlated with the image. Performing the task of cross-correlation with the flipped filter is typically called convolution. Therefore, at any particular pixel location, the convolution may be mathematically described by

$$H_{dig}(u, v) = \sum_{i=-n}^n \sum_{j=-m}^m G(i + n + 1, j + m + 1) I_{dig}(u - i, v - j) \quad (4.11)$$

Applying this relation to all (u, v) pixel locations,

$$H_{dig} = G * I_{dig} \quad (4.12)$$

where $*$ denotes the convolution operator and H_{dig} denotes the filtered image. In the above, it is said that G has been convolved with I_{dig} to yield H_{dig}

The values of the kernel are usually normalized to sum to 1 so that the convolution (or correlation) process does not brighten or dim the image. There are many standard kernels that are frequently used by the image processing community. Two kernels worth mentioning here are the Gaussian kernel and the Laplacian of the Gaussian kernel, both shown graphically in Fig. 4.7. In subsequent sections, the Gaussian kernel will be used as a template for the star finding algorithm and to simulate image blurring. The Laplacian of the Gaussian kernel is a standard edge detection filter.

The cross-correlation operation (or the convolution operation) may also be thought of as a dot product. If the columns of G are stacked on top of each other to form the column vector \mathbf{g} , and the columns of the image patch are stacked on top of each other to form the column vector \mathbf{p} , then the cross-

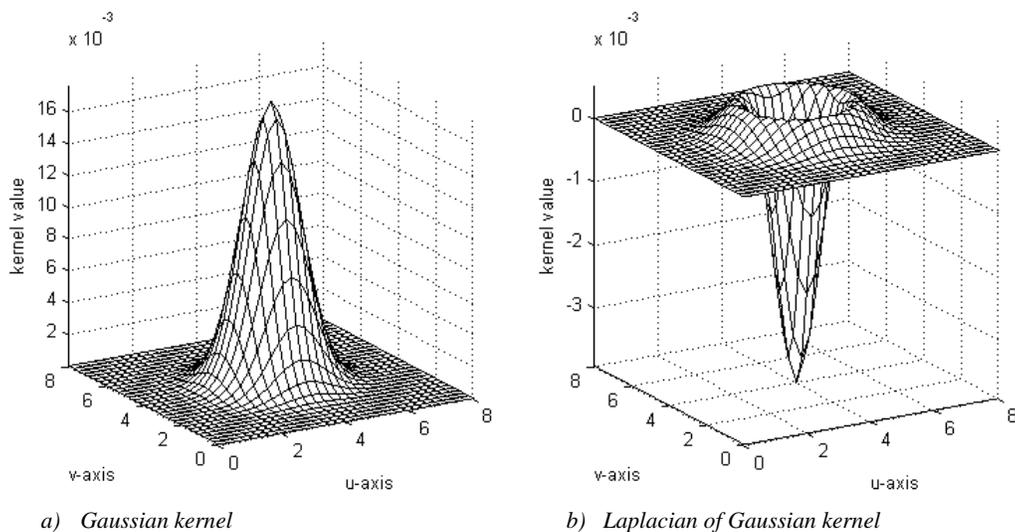


Figure 4.7: Three dimensional visualization of important kernel shapes.

correlation function of Eq. 4.9 may be written as

$$J_{dig}(u, v) = \mathbf{g}^T \mathbf{p} \quad (4.13)$$

This dot product will be a maximum when \mathbf{g} and \mathbf{p} are in the same direction. This means that the maximum response of a cross-correlation will be observed when the patch from the original image “looks like” the filter. Therefore, filters may be used as templates. To prevent brighter portions of the image from registering a higher response simply due to their high intensity values, the image patch vector may be normalized at each (u, v) location,

$$J'_{dig}(u, v) = \frac{\mathbf{g}^T \mathbf{p}}{\|\mathbf{p}\|} \quad (4.14)$$

This process, typically called normalized correlation, allows for template matching independent of image brightness. Note that the values in the kernel may also be normalized such that the filter column vector \mathbf{g} is also a unit vector. Under this scenario, Eq. 4.14 is simply the dot product between two unit vectors that will be maximized when the pixel intensity pattern in the kernel and the image patch are the same.

4.2.2 Thresholding

The technique of thresholding is used to convert a n -bit digital image (or filter output), which usually has pixels that can take on integer intensity values from 0 to $2^n - 1$, to a binary image. The pixels in a binary image may only take on two values. A binary pixel value of 1 will indicate a white pixel, while a binary pixel value of 0 will indicate a black pixel.

A threshold algorithm is easy to construct. Suppose a threshold value of τ is selected. Creating a thresholded image requires a check at each pixel location, where each pixel with intensity above τ is set to 1 (white) and each pixel with intensity below τ is set to 0 (black). Those pixels with a value of 1 are said to be in the “foreground” and those pixels with a value of 0 are said to be in the “background.”

To demonstrate the effect of thresholding, example OPNAV images from the MESSENGER spacecraft’s Wide Angle Camera (WAC) are used. The threshold operation is performed on an 8 bit version of these images. Therefore, intensity values at each pixel can range from 0 to 255. Results of thresholding and image from the August 2005 Earth flyby are shown in Fig. 4.8 and results from the January 2008 Mercury flyby are shown in Fig. 4.9.

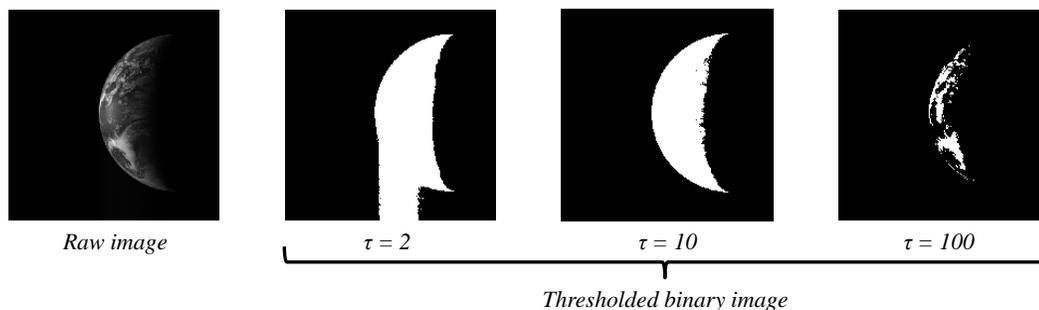


Figure 4.8: Raw and thresholded images of Earth. The raw image was taken by the MESSENGER wide angle camera on 2 August 2005 during the spacecraft’s Earth flyby (Product ID EW0031513371D from [93]).

4.2.3 Dilation and Erosion

Given a binary image, the process of dilation adds additional foreground pixels around existing foreground pixels. Dilation will cause the number of pixels around existing foreground pixels. Dilation will cause the number of foreground pixels to grow and will eliminate small areas of background pixels surrounded by foreground pixels. This procedure for dilating an image is straightforward. Begin by introducing the idea of a structuring matrix, \mathbf{S} , which is a $n \times m$ matrix of 0s and 1s. Then at each pixel location in the original binary image, the structuring matrix is centered on the pixel of interest. The pixels in the original image that correspond to the elements of \mathbf{S} that have a value of 1 are called the neighborhood of the current pixel. If any pixel in the neighborhood of the current pixel is foreground, then the current pixel is set to foreground.

Erosion is the opposite of dilation. Erosion will remove foreground pixels that are near existing background pixels. This process will cause the

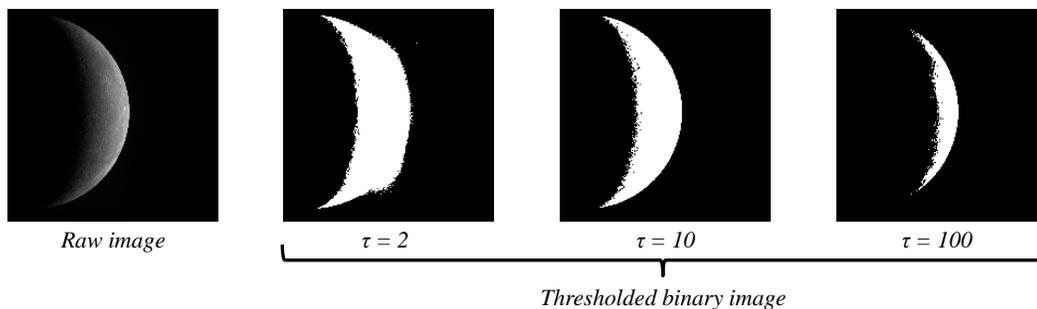


Figure 4.9: Raw and thresholded images of Mercury. The raw image was taken by the MESSENGER wide angle camera on 14 January 2008 during the spacecraft’s Mercury flyby (Product ID EW0108820032H from [93]).

number of foreground pixels to shrink and will eliminate small areas of foreground pixels surrounded by background pixels. As with dilation, a structuring matrix is used to add or remove foreground pixels. In erosion, if any pixel in the neighborhood of the current pixel is background, then the current pixel is set to background.

The process of “opening” an image consists of eroding, and then dilating the image. This will remove small (or thin) foreground objects, while maintaining the original shape of larger foreground objects. Although opening a thresholded OPNAV image may be used to eliminate isolated instances of blooming (if a CCD detector is used), it will probably remove all the stars in the image.

The process of “closing” an image consists of dilating, and then eroding the image. This will fill in small background areas that are surrounded by foreground, while maintaining the original shape of the foreground objects. This may help connect foreground areas in a heavily cratered planet/moon that are separated by areas of shadow.

An example of opening and closing on an real OPNAV image from the MESSENGER spacecraft may be seen in Fig. 4.10. The structuring matrix is a circle of ones with a radius of 7 pixels.

4.2.4 Connected Components

Connected components analysis is a grouping algorithm for binary images. In this technique, foreground pixels that are connected to one another

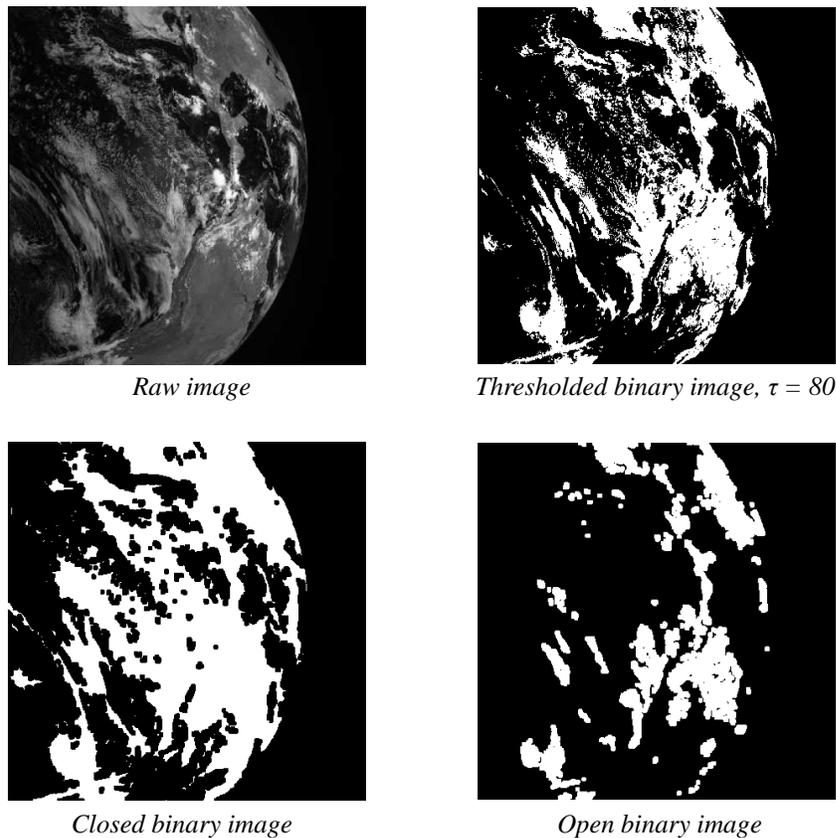


Figure 4.10: Dilation and erosion of an OPNAV image of Earth. The raw image was taken by the MESSENGER wide angle camera on 2 August 2005 during the spacecraft’s Earth flyby (Product ID EW0031489478I from [93]).

are considered to belong to the same group. Pixel connectivity is usually determined by either using a four-connected or eight-connected schema as shown in Fig. 4.11.

As an example, the connected components algorithm (eight-connected) is applied to the thresholded, opened, and closed OPNAV image from Fig. 4.10. The results of the connected components analysis are shown in Fig.

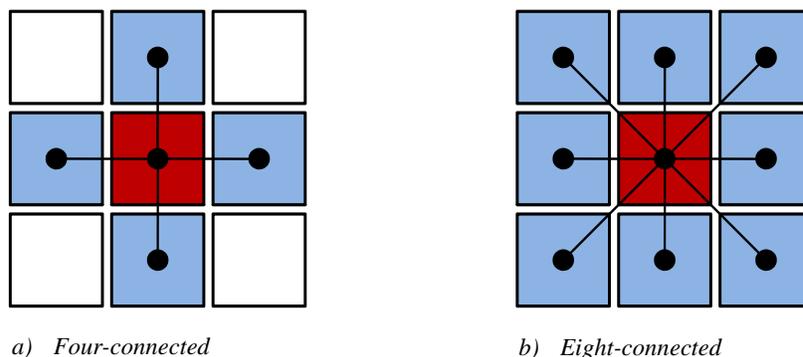


Figure 4.11: Graphical explanation of a pixel that is (a) four-connected and (b) eight-connected. Pixels in blue are considered to be connected to the red center pixel.

4.12. The different groups in Fig. 4.12 are color coded, although colors repeat in some cases because there are more groups than available colors. Of particular note is how closing an image may be used to connect regions that were previously not connected due to differences in albedo and the threshold choice.

4.3 Computing Distance to an Ellipse

Computing the distance from a point to an ellipse is a fundamental task important to many of the algorithms developed in this chapter. The distance to an ellipse is necessary to compute the residuals when fitting an elliptical model to a set of noisy data points. It is also necessary to compute the distance from a reference star to an elliptical estimate of the horizon location.

This section begins with a discussion of ellipse representations that will be important in this and future discussions. It is followed by a discussion of

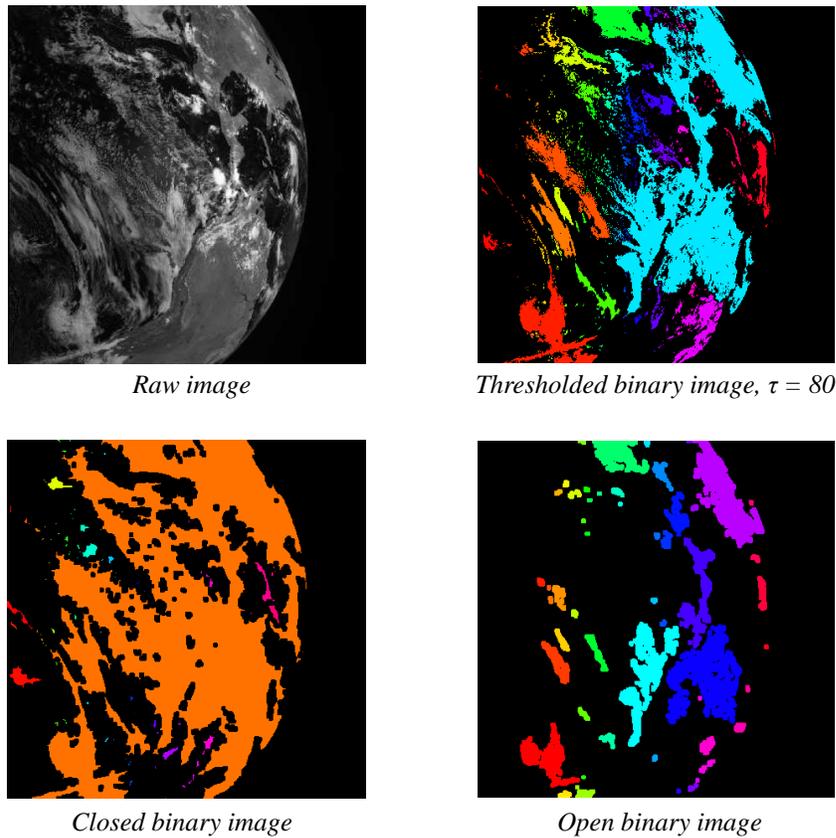


Figure 4.12: Connected components analysis of an OPNAV image of Earth from Fig. 4.10. Each group of components is indicated by a different color.

computing geometric distance and algebraic distance.

4.3.1 Representations of an Ellipse

4.3.1.1 Implicit Equation for Ellipse

Any conic section may be described by the following implicit quadratic equation:[134]

$$F(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \quad (4.15)$$

where the point $[x, y]$ lies on the conic section. A detailed discussion of the properties of this implicit equation is given in [134] and it is well known that the conic will be an ellipse if $B^2 - 4AC < 0$.

4.3.1.2 Standard Ellipse Parameters

The coefficients of the implicit conic equation are not the most convenient means of describing an ellipse for many applications. Instead it is frequently easier to describe an ellipse in terms of the following five parameters: semimajor axis, a ; the semiminor axis, b ; coordinate of the ellipse center, $[x_0, y_0]$; and the angle from the x-axis to the ellipse major axis. The transformation from the implicit coefficients to these standard ellipse parameters is a well-known result from classical geometry and is given by:

$$x_0 = \frac{2CD - BE}{B^2 - 4AC} \quad y_0 = \frac{2AE - BD}{B^2 - 4AC} \quad (4.16)$$

$$a = \frac{\sqrt{2[AE^2 + CD^2 - BDE + F(B^2 - 4AC)]}}{\sqrt{(B^2 - 4AC) \left[\sqrt{(A - C)^2 + B^2} - A - C \right]}} \quad (4.17)$$

$$b = \frac{\sqrt{2[AE^2 + CD^2 - BDE + F(B^2 - 4AC)]}}{\sqrt{(B^2 - 4AC) \left[-\sqrt{(A - C)^2 + B^2} - A - C \right]}} \quad (4.18)$$

$$\phi = \begin{cases} 0 & B = 0 \text{ and } A < C \\ \frac{\pi}{2} & B = 0 \text{ and } A > C \\ \frac{1}{2} \cot^{-1} \left(\frac{A-C}{B} \right) & B \neq 0 \text{ and } A < C \\ \frac{\pi}{2} + \frac{1}{2} \cot^{-1} \left(\frac{A-C}{B} \right) & B \neq 0 \text{ and } A > C \end{cases} \quad (4.19)$$

4.3.1.3 Matrix Representation

In many cases, it may be useful to represent an ellipse in a matrix quadratic form. Two such forms are useful. The first is a quadratic form in terms of the implicit conic equation and homogeneous coordinates (see section 3.1.2). It is straightforward to verify that Eq 4.15 may be rewritten as

$$F(x, y) = \mathbf{x}_h^T \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix} \mathbf{x}_h \quad (4.20)$$

where $\mathbf{x}_h = [x \ y \ 1]^T$.

The second important matrix quadratic form is in terms of the standard ellipse parameters. The most basic description of an ellipse that is centered at the origin and aligned with its principal axes is given by

$$\frac{x'}{a^2} + \frac{y'}{b^2} = \mathbf{x}'^T \begin{bmatrix} 1/a^2 & 0 \\ 0 & 1/b^2 \end{bmatrix} \mathbf{x}' = \mathbf{x}'^T \mathbf{A}' \mathbf{x}' = 1 \quad (4.21)$$

where $\mathbf{x}' = [x' \ y']^T$ and the primes denote the coordinates expressed in the ellipse principal axis frame. Therefore, an ellipse with a center at $[x_0, y_0]$ and with a semimajor axis rotated by an angle ϕ with respect to the x-axis is given by

$$[\mathbf{x} - \mathbf{x}_0]^T \mathbf{T}_\phi^T \begin{bmatrix} 1/a^2 & 0 \\ 0 & 1/b^2 \end{bmatrix} \mathbf{T}_\phi [\mathbf{x} - \mathbf{x}_0] = 1 \quad (4.22)$$

$$[\mathbf{x} - \mathbf{x}_0]^T \mathbf{A} [\mathbf{x} - \mathbf{x}_0] = 1 \quad (4.23)$$

where $\mathbf{x}_0 = [x_0 \ y_0]^T$ and the matrix \mathbf{T}_ϕ rotates a point from the original coordinate frame to a frame aligned with the principal axes of the ellipse,

$$\mathbf{T}_\phi = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \quad (4.24)$$

Therefore, the matrix \mathbf{A} that describes an ellipse with a general orientation may be diagonalized by

$$\mathbf{A} = \mathbf{T}_\phi^T \mathbf{A}' \mathbf{T}_\phi \quad (4.25)$$

4.3.2 Geometric Distance

Define the geometric distance from the point $[x, y]$ to an ellipse as Euclidean distance from that point to the closest point on the ellipse, $[x_e, y_e]$. The line that connects the point $[x, y]$ to $[x_e, y_e]$ should be perpendicular to the tangent of the ellipse at $[x_e, y_e]$. Unfortunately, a closed-form analytic solution to this problem is not available and most modern techniques rely on a numerical approach. The approach used here to solve this problem is derived below.

To simplify the problem, begin by shifting the origin to the center of the ellipse, $[x_0, y_0]$. Let these transformed coordinates be represented by

$$\mathbf{z} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad \mathbf{z}_e = \begin{bmatrix} x_e \\ y_e \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (4.26)$$

The quantities \mathbf{z} and \mathbf{z}_e may also be expressed in the ellipse principal axis frame as follows,

$$\mathbf{z}' = \mathbf{T}_\phi \mathbf{z} \quad \mathbf{z}'_e = \mathbf{T}_\phi \mathbf{z}_e \quad (4.27)$$

where, as before, the primed variables are expressed in the ellipse principal axis frame.

Now, the geometric distance problem may be stated as the following

optimization problem

$$\min J(\mathbf{z}_e) = \|\mathbf{z} - \mathbf{z}_e\|^2 \quad (4.28)$$

subject to (from Eq. 4.23)

$$\mathbf{z}_e^T \mathbf{A} \mathbf{z}_e = 1 \quad (4.29)$$

This optimization problem is the same regardless of the frame. Therefore, the following objective function will produce the same solution

$$\min J(\mathbf{z}'_e) = \|\mathbf{z}' - \mathbf{z}'_e\|^2 \quad (4.30)$$

$$(\mathbf{z}'_e)^T \mathbf{A}' \mathbf{z}'_e = 1 \quad (4.31)$$

Using the form of the objective function in the ellipse principal axis frame, adjoin the constraint to the objective function with a Lagrange multiplier

$$\min \mathcal{J}(\mathbf{z}'_e) = (\mathbf{z}')^T \mathbf{z}' - 2(\mathbf{z}')^T \mathbf{z}'_e + (\mathbf{z}'_e)^T \mathbf{z}'_e + \lambda \left[(\mathbf{z}'_e)^T \mathbf{A}' \mathbf{z}'_e - 1 \right] \quad (4.32)$$

Recognizing that a minimum exists when the first differential is zero, it is straightforward to show that a minimum occurs at

$$-2(\mathbf{z}')^T + 2(\mathbf{z}'_e)^T + 2\lambda(\mathbf{z}'_e)^T \mathbf{A}' = \mathbf{0}_{2 \times 1} \quad (4.33)$$

$$(\mathbf{z}'_e)^T \mathbf{A}' \mathbf{z}'_e = 1 \quad (4.34)$$

Recognizing that \mathbf{A}' is symmetric, Eq. 4.33 may be solved for \mathbf{z}'_e ,

$$\mathbf{z}'_e = [\mathbf{I}_{2 \times 2} + \lambda \mathbf{A}']^{-1} \mathbf{z}' \quad (4.35)$$

Further, it may be shown that this relation is also true in the unrotated frame,

$$\mathbf{z}_e = [\mathbf{I}_{2 \times 2} + \lambda \mathbf{A}]^{-1} \mathbf{z} \quad (4.36)$$

where the Lagrange multiplier λ is the same in both frames. Proceed by substituting Eq. 4.35 into Eq. 4.34,

$$(\mathbf{z}')^T [\mathbf{I}_{2 \times 2} + \lambda \mathbf{A}']^{-T} \mathbf{A}' [\mathbf{I}_{2 \times 2} + \lambda \mathbf{A}']^{-1} \mathbf{z}' = 1 \quad (4.37)$$

Recalling the identity from Eq. 4.27, this may be further rewritten as

$$\mathbf{z}^T \mathbf{T}_\phi^T [\mathbf{I}_{2 \times 2} + \lambda \mathbf{A}']^{-1} \mathbf{A}' [\mathbf{I}_{2 \times 2} + \lambda \mathbf{A}']^{-1} \mathbf{T}_\phi \mathbf{z} = 1 \quad (4.38)$$

Substituting for \mathbf{A}' from Eq. 4.21 and \mathbf{T}_ϕ from Eq. 4.24, allows this 2×2 matrix equation to be expanded explicitly as

$$\frac{a^2}{(a^2 + \lambda)^2} (z_1 \cos \phi + z_2 \sin \phi)^2 + \frac{b^2}{(b^2 + \lambda)^2} (-z_1 \sin \phi + z_2 \cos \phi)^2 = 1 \quad (4.39)$$

where $\mathbf{z} = [z_1 \ z_2]^T$. This, of course, can be rewritten as a fourth-order equation in λ ,

$$g(\lambda) = C_4 \lambda^4 + C_3 \lambda^3 + C_2 \lambda^2 + C_1 \lambda + C_0 = 0 \quad (4.40)$$

where

$$C_0 = a^4 b^4 - G_2 a^4 - G_1 b^4 \quad (4.41)$$

$$C_1 = 2(b^4 - G_2) a^2 + 2(a^4 - G_1) b^2 \quad (4.42)$$

$$C_2 = a^4 + a^2 b^2 + b^4 - G_1 - G_2 \quad (4.43)$$

$$C_3 = 2(a^2 + b^2) \quad C_4 = 1 \quad (4.44)$$

and

$$G_1 = a^2 (z_1 \cos \phi + z_2 \sin \phi)^2 \quad G_2 = b^2 (-z_1 \sin \phi + z_2 \cos \phi)^2 \quad (4.45)$$

Solving Eq. 4.40 will yield four possible values of λ . The solution implemented in this dissertation used the ‘roots’ command in MATLAB to perform this task. The only remaining difficulty is choosing the right root of $g(\lambda)$. Looking at the second differential condition, which states that a minimum occurs when $d^2\mathcal{J}(\mathbf{z}'_e)/d\mathbf{z}'_e d\mathbf{z}'_e$ is positive definite,

$$\frac{d^2\mathcal{J}(\mathbf{z}'_e)}{d\mathbf{z}'_e d\mathbf{z}'_e} = \mathbf{I}_{2 \times 2} + \lambda \mathbf{A}' > 0 \quad (4.46)$$

Because the matrix in Eq. 4.46 is diagonal, this is the same as

$$1 + \frac{\lambda}{a^2} > 0 \quad 1 + \frac{\lambda}{b^2} > 0 \quad (4.47)$$

These conditions will only be met by one value of λ . At least one other real value of λ will exist and will correspond to a point on the other side of the ellipse that is the furthest point on the ellipse from $[x, y]$ (see Fig 4.13). Once λ is known, Eq. 4.36 may be used to find \mathbf{z}_e and Eq. 4.26 may be used to transform \mathbf{z}_e back to the original coordinates to obtain $[x_e, y_e]$.

4.3.3 Algebraic Distance

Define the algebraic distance of a point $[x, y]$ from a conic section to be the value of the implicit quadratic equation, $F(x, y)$, from Eq. 4.15. Any point on the ellipse has a distance of zero, i.e. $F(x, y) = 0$ if $[x, y]$ is on the conic section. As a point moves farther from the conic section, the value of $F(x, y)$ increases.

It is important to note that geometric distance and algebraic distance are not the same and that they differ by more than just a simple scale factor.

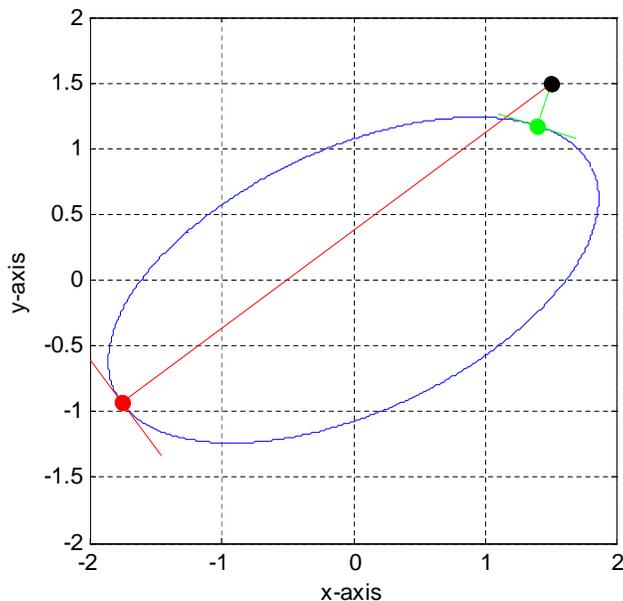


Figure 4.13: Graphical example of the ellipse points generated by the real roots of Eq. 4.40. One point represents the closest point on the ellipse (green) and the other point represents the furthest point on the ellipse (red). Example ellipse is centered at the origin with $a = 2$, $b = 1$, and $\phi = 25$ degrees.

Two different points with the same geometric distance from the ellipse will have different algebraic distances, depending on their location around the ellipse. The example in Fig. 4.14 provides a graphical comparison of geometric distance and algebraic distance.

4.4 Ellipse Fitting

The perspective projection of a sphere or ellipsoid will form an ellipse on the image plane. Therefore, if the objective is to locate a planet or moon in an image, the image must be searched for an ellipse. The discussion that

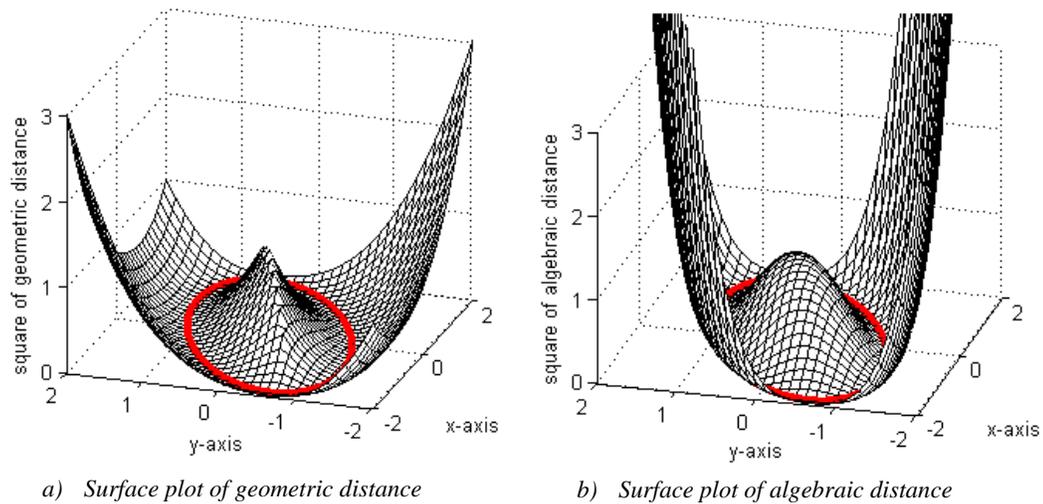


Figure 4.14: Comparison of (a) geometric distance and (b) algebraic distance from an example ellipse (red) that is centered at the origin with $a = 2$, $b = 1$, and $\phi = 25$ degrees.

follows assumes that measured data points on the edge of the ellipse have already been detected. The issue of how these points are found is discussed later (see Section 4.7). Once candidate edge points are found, an ellipse must be fit to this data set. The details of this ellipse fitting process are discussed in the following sections.

4.4.1 Basic Ellipse Fitting with Direct Least Squares Estimation

Most early solutions to the problem of ellipse fitting were based on clustering and voting techniques, such as the Hough transform. While these techniques are naturally robust to noise and outliers, they require large amounts of memory and time-consuming computations. This has lead many to consider least squares based approaches.[135] Although least squares solutions are much

more sensitive to the presence of outliers, the robust model fitting techniques described in Section 4.5 may be used to solve this drawback separately.

The first efficient approach for ellipse fitting was a direct least squares algorithm developed by Fitzgibbon, Pilum and Fisher.[136]¹ This basic approach will provide the theoretical underpinnings for the improved ellipse fitting technique described in the next section.

Begin by recalling that any conic section may be described by an equation of the form

$$F(\mathbf{a}, \mathbf{x}_i) = \mathbf{a}^T \mathbf{x}_i = Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F = 0 \quad (4.48)$$

where $[x_i, y_i]$ is a point on the conic section, $\mathbf{a} = [A \ B \ C \ D \ E \ F]^T$, and $\mathbf{x}_i = [x_i^2 \ x_iy_i \ y_i^2 \ x_i \ y_i \ 1]^T$. The conic is an ellipse if $B^2 - 4AC < 0$. The form of Eq. 4.48 allows for the constants to be arbitrarily scaled such that the ellipse inequality constraint may be rewritten as an equality constraint,

$$4AC - B^2 = 1 \quad (4.49)$$

In general, noise in the data will cause any point $[x, y]$ to not lie exactly on the ellipse and $F(\mathbf{a}, \mathbf{x}_i) \neq 0$. Therefore, the following optimization problem based on the square of the model fit residuals (expressed in algebraic distance) is proposed [136]:

$$\min J = \sum_{i=1}^n [F(\mathbf{a}, \mathbf{x}_i)]^2 = \mathbf{a}^T \mathbf{D}^T \mathbf{D} \mathbf{a} \quad (4.50)$$

¹Although [136] was published in a journal in 1999, the authors first presented these findings at a conference in 1996.

where $D = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_i]^T$, subject to the equality constraint of Eq. 4.49 rewritten in vector form,

$$4AC - B^2 = \mathbf{a}^T \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{a} = \mathbf{a}^T \mathbf{C} \mathbf{a} = 1 \quad (4.51)$$

If the equality constraint of Eq. 4.51 is adjoined to the objective function with a Lagrange multiplier, then

$$\min \quad \mathcal{J} = \mathbf{a}^T \mathbf{D}^T \mathbf{D} \mathbf{a} + \lambda (1 - \mathbf{a}^T \mathbf{C} \mathbf{a}) \quad (4.52)$$

From here, it is easy to show that the solution to this optimization problem is the following rank-deficient generalized eigenvalue problem

$$(\mathbf{D}^T \mathbf{D}) \mathbf{a} = \lambda \mathbf{C} \mathbf{a} \quad (4.53)$$

Substituting this solution into the objective function shows that

$$J = \mathbf{a}^T \mathbf{D}^T \mathbf{D} \mathbf{a} = \lambda \mathbf{a}^T \mathbf{C} \mathbf{a} = \lambda \quad (4.54)$$

meaning that the optimal solution occurs at the minimal positive eigenvalue. It was shown in [136] that the solution to Eq. 4.53 yields exactly one positive eigenvalue. This positive eigenvalue corresponds to the eigenvector that is the solution for \mathbf{a} .

4.4.2 Improved Ellipse Fitting with Direct Least Squares Estimation

Despite a clean theoretical derivation, Fitzgibbon's approach for fitting an ellipse to measured edge points (described in the previous section) suffers

from a number of practical difficulties. The direct implementation of Fitzgibbon's approach by simply solving the eigenvalue problem of Eq. 4.53 leads to numerous sources of numerical instability. The two most important difficulties are:

1. Fitzgibbon's approach only works on noisy data. If the measured points lie exactly on an ellipse (or very close to on an ellipse), then the matrix $\mathbf{D}^T\mathbf{D}$ becomes singular and no solution is obtained. This is an artifact of the construction of Eq. 4.53 and not an inherent weakness of this approach.
2. For very high-resolution images, the pixel locations may become very large relative to the size of the ellipse (e.g. a relatively small ellipse in the bottom left-hand corner of the image). This can lead to the matrix $\mathbf{D}^T\mathbf{D}$ becoming ill-conditioned.

The first difficulty, the issue of $\mathbf{D}^T\mathbf{D}$ being singular when the edge points fall exactly on (or very nearly on) an ellipse, was overcome by the work of Halíř and Flusser in 1998.[137] This technique exploits the structure of the \mathbf{C} and \mathbf{D} matrices to simplify the eigenvalue problem of Eq. 4.53. Begin by partitioning \mathbf{a} , \mathbf{C} , and \mathbf{D} as follows:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} \quad \mathbf{a}_1 = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \quad \mathbf{a}_2 = \begin{bmatrix} D \\ E \\ F \end{bmatrix} \quad (4.55)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad \mathbf{C}_1 = \begin{bmatrix} 0 & 0 & 2 \\ 0 & -1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \quad (4.56)$$

$$\mathbf{D} = [\mathbf{D}_1 \quad \mathbf{D}_2] \quad (4.57)$$

$$\mathbf{D}_1 = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 \\ x_2^2 & x_2 y_2 & y_2^2 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n y_n & y_n^2 \end{bmatrix} \quad \mathbf{D}_2 = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \quad (4.58)$$

To compact notation, define the scatter matrix as $\mathbf{S} = \mathbf{D}^T \mathbf{D}$ such that

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{S}_2^T & \mathbf{S}_3 \end{bmatrix} \quad (4.59)$$

and

$$\mathbf{S}_1 = \mathbf{D}_1^T \mathbf{D}_1 \quad \mathbf{S}_2 = \mathbf{D}_1^T \mathbf{D}_2 \quad \mathbf{S}_3 = \mathbf{D}_2^T \mathbf{D}_2 \quad (4.60)$$

Inserting these partitioned matrices into Eq. 4.53, it is straightforward to show that the solution reduces to the following two equations

$$\mathbf{S}_1 \mathbf{a}_1 + \mathbf{S}_2 \mathbf{a}_2 = \lambda \mathbf{C}_1 \mathbf{a}_1 \quad (4.61)$$

$$\mathbf{S}_2^T \mathbf{a}_1 + \mathbf{S}_3 \mathbf{a}_2 = \mathbf{0}_{3 \times 1} \quad (4.62)$$

Because the matrix \mathbf{S}_3 is only singular if all the points lie on a line (a case where no elliptical fit would be possible), the matrix \mathbf{S}_3 is generally invertible in practice and \mathbf{a}_2 may be found by

$$\mathbf{a}_2 = -\mathbf{S}_3^{-1} \mathbf{S}_2^T \mathbf{a}_1 \quad (4.63)$$

The only remaining task is to solve for \mathbf{a}_1 . Therefore, inserting Eq. 4.63 into Eq. 4.61 and noting that the matrix \mathbf{C}_1 is invertible, the solution for \mathbf{a}_1 is given by a simple 3×3 eigenvalue problem,

$$\mathbf{M} \mathbf{a}_1 = \lambda \mathbf{a}_1 \quad (4.64)$$

where

$$\mathbf{M} = \mathbf{C}_1^{-1} [\mathbf{S}_1 - \mathbf{S}_2 \mathbf{S}_3^{-1} \mathbf{S}_2^T] \quad (4.65)$$

The solution to Eq. 4.64 will produce three possible solutions. It was shown in [136], however, that only one elliptical solution may exist. Therefore, the ellipse condition of $4AC - B^2 > 0$ is used to select the appropriate eigenvector. This condition will be met for one, and only one, of the three eigenvectors of \mathbf{M} . This eigenvector is the solution for \mathbf{a}_1 .

The second difficulty is an issue of centering and scaling. Suppose that for the purposes of ellipse fitting, the origin of the coordinate systems is temporarily shifted to the mean $[x, y]$ location,

$$x_c = \frac{1}{n} \sum_{i=1}^n x_i \quad y_c = \frac{1}{n} \sum_{i=1}^n y_i \quad (4.66)$$

where x_c and y_c are the coordinates of the new coordinate system origin. Additionally, suppose the axes are rescaled such that

$$\tilde{x} = \frac{x - x_c}{s} \quad \tilde{y} = \frac{y - y_c}{s} \quad (4.67)$$

where $[\tilde{x}, \tilde{y}]$ are the new coordinates of each data point and s is the scale factor. Let the scale factor be chosen based on the largest dimension,

$$s = \frac{1}{2} \max \left\{ \begin{array}{l} \max \{x_i\} - \min \{x_i\} \\ \max \{y_i\} - \min \{y_i\} \end{array} \right\} \quad (4.68)$$

The solution to Eq. 4.63 and Eq. 4.64 will yield the coefficients of the implicit conic equation in the transformed coordinates, $\tilde{\mathbf{a}} = [\tilde{A} \tilde{B} \tilde{C} \tilde{D} \tilde{E} \tilde{F}]^T$.

Recognizing that a point on the ellipse must satisfy the implicit conic equation in both sets of coordinates

$$\mathbf{a}^T \mathbf{x}_i = \tilde{\mathbf{a}}^T \tilde{\mathbf{x}}_i = 0 \quad (4.69)$$

it may be shown that the following transformation is true²

$$A = \tilde{A} \quad B = \tilde{B} \quad C = \tilde{C} \quad (4.70)$$

$$D = \left(\tilde{D}s - \tilde{B}y_c - 2\tilde{A}x_c \right) \quad (4.71)$$

$$E = \left(\tilde{E}s - \tilde{B}x_c - 2\tilde{C}y_c \right) \quad (4.72)$$

$$F = \left(\tilde{A}x_c^2 + \tilde{B}x_c y_c + \tilde{C}y_c^2 - \tilde{D}x_c s - \tilde{E}y_c s + \tilde{F}s^2 \right) \quad (4.73)$$

4.5 Robust Model Fitting

The robust fitting of a model to noisy data is an area of active research in the computer vision community. The most widely used approach to solving this problem in modern systems is the RANdom SAmple Consensus (RANSAC) algorithm. Numerous important improvements have been made to the classic RANSAC algorithm. Of particular importance here is the M-Estimator SAmple Consensus (MSAC) algorithm. Both the RANSAC and MSAC algorithms are discussed in the following sections.

²This approach of using $\mathbf{a}^T \mathbf{x}_i = \tilde{\mathbf{a}}^T \tilde{\mathbf{x}}_i = 0$ to transform between two sets of coefficients for the implicit conic equation was originally shown in [138]. Unfortunately, the original derivation in [138] contains a few minor mistakes that are corrected in the transformation shown here. Note that [138] also uses a different scaling factor for the x -direction and y -direction, instead of using the same scaling factor as in Eq. 4.67.

4.5.1 RANdOm SAmple Consensus (RANSAC)

The RANSAC algorithm, developed by Fischler and Bolles in 1981,[139] is an iterative method used for the robust fitting of a model to a set of noisy data. Rather than trying to find and eliminate outliers in the data set, the RANSAC algorithm uses a small subset of the data to fit the model and then looks for “inliers” (or the number remaining data points that are close to the model). This allows for the model fitting to occur in an extremely noisy environment.

The basic RANSAC algorithm consists of the following procedure:

1. Select the type of model to fit to the data (e.g. line, circle, ellipse, etc.)
2. Perform the following tasks k times:
 - (a) Randomly select n points from the data set
 - (b) Fit the model to these n data points
 - (c) For each data point, check to see if distance from the point to the model is less than t . If the distance is less than t , then the point is said to be “close” to the model.
3. Of the k models generated in Step 2, select the model with the largest number of close data points.

For a data set of fixed size, maximizing the number of close points is the same as minimizing the number of outliers. Therefore, the RANSAC algorithm

may be described mathematically by the following optimization problem

$$\min \quad J(\mathbf{p}) = \sum_{i=1}^n \rho(\mathbf{x}_i, \mathbf{p}) \quad (4.74)$$

where \mathbf{p} is a vector of parameters that describes the model, $\mathbf{x}_i = [x_i \ y_i]^T$, and ρ is defined as

$$\rho(\mathbf{x}_i, \mathbf{p}) = \begin{cases} 0 & d(\mathbf{x}_i, \mathbf{p})^2 < t^2 \\ 1 & d(\mathbf{x}_i, \mathbf{p})^2 \geq t^2 \end{cases} \quad (4.75)$$

The term $d(\mathbf{x}_i, \mathbf{p})$ is the distance between the point $[x_i, y_i]$ and the model described by \mathbf{p} .

It is worth highlighting that the random selection of the n points used to fit the model at each of the k iterations means that RANSAC algorithm is not deterministic. Therefore, given the same set of data, the best fit solution generated by RANSAC will not necessarily be the same every time. Clearly, the choice of the number of data points to fit, n , the number of iterations, k , and the error threshold used to define “inliers,” t , all affect the performance (both quality of the model fit and algorithm run-time) of the RANSAC algorithm in practice. There are many different approaches - some heuristic and some based on theory - that may be used to pick n , k , and t . Some various approaches are discussed in [87].

As an example, consider the problem of fitting a line to noisy data with a large number of outliers (see Fig. 4.15). The outliers significantly affect the results of the least squares approach, but the RANSAC algorithm is capable of detecting the outliers and provides a good estimate of the true model even in a high noise environment.

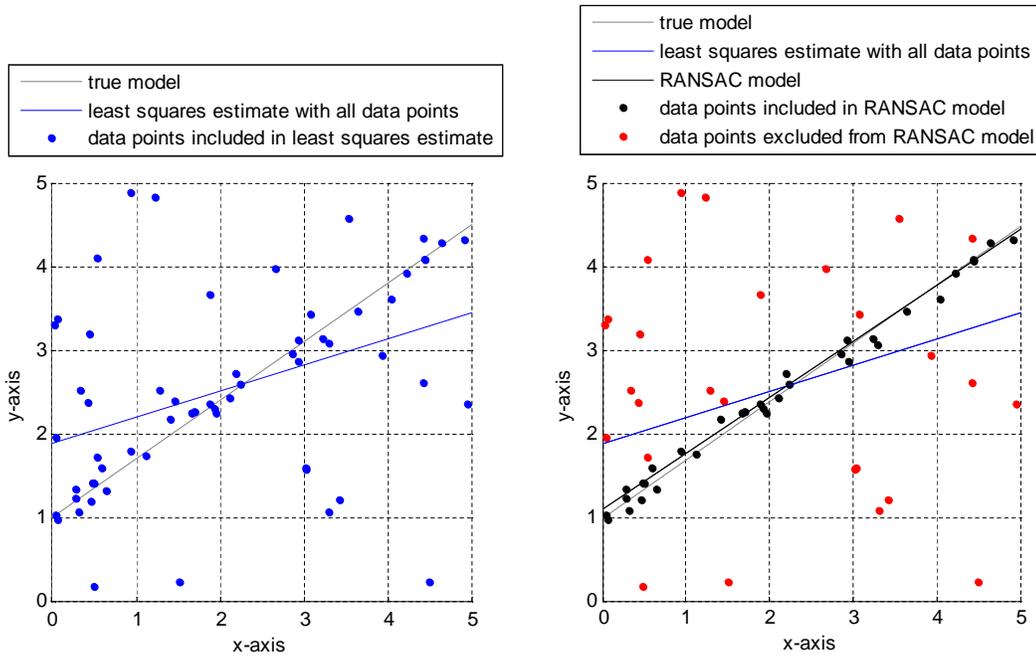


Figure 4.15: Example of RANSAC algorithm applied to noisy data along a line with a large number of outliers.

4.5.2 M-Estimator Sample Consensus (MSAC)

The MSAC algorithm, developed by Torr and Zisserman,[140, 141] uses an M-estimation approach to improve the RANSAC algorithm introduced in the previous section. A detailed discussion of M-estimators may be found in [142] and [143].

In general, an M-estimator minimizes the following objective function

$$\min_{\mathbf{p}} J(\mathbf{p}) = \sum_{i=1}^n \rho(\mathbf{x}_i, \mathbf{p}) \quad (4.76)$$

where \mathbf{p} is a vector of parameters that describes the model, \mathbf{x}_i is a vector of the i -th set of measurements, and $\rho(\mathbf{x}_i, \mathbf{p})$ is a symmetric, positive-definite

function with a unique minimum at zero. The RANSAC objective function, with $\rho(\mathbf{x}_i, \mathbf{p})$ from Eq. 4.75, is a special case of an M-estimator. As before, define $d(\mathbf{x}_i, \mathbf{p})$ as the distance (or residual) between the model and the measured data. It is interesting to note that if $\rho(\mathbf{x}_i, \mathbf{p}) = d(\mathbf{x}_i, \mathbf{p})^2/2$ then the result is a least-squares estimate. Therefore, the least-squares solution is also a special case of an M-estimator.

Looking at the form of Eq. 4.75 in RANSAC, it was proposed in [140] and [141] to replace $\rho(\mathbf{x}_i, \mathbf{p})$ with the following expression to create the MSAC algorithm:

$$\rho(\mathbf{x}_i, \mathbf{p}) = \begin{cases} d(\mathbf{x}_i, \mathbf{p})^2 & d(\mathbf{x}_i, \mathbf{p})^2 < t^2 \\ t^2 & d(\mathbf{x}_i, \mathbf{p})^2 \geq t^2 \end{cases} \quad (4.77)$$

The form of Eq. 4.77 is simply the Huber M-estimator function. The MSAC algorithm has been shown to provide better results than RANSAC with no additional computational burden. Therefore, the MSAC algorithm is given by the following procedure:

1. Select the type of model to fit to the data (e.g. line, circle, ellipse, etc.)
2. Perform the following tasks k times:
 - (a) Randomly select n points from the data set
 - (b) Fit the model to these n data points
 - (c) Compute $J(\mathbf{p})$ from Eq. 4.76 and Eq. 4.77 for each model.
3. Of the k models generated in Step 2, select the model with the smallest value for $J(\mathbf{p})$.

4.6 Identifying the Location of Candidate Stars

Because stars will appear as blurred point sources, there are no significant issues associated with scaling and candidate stars may be found by simple template matching. It is worth noting, however, that smearing of the stars in an image can occur if the exposure time is long and the vehicle is rotating. This effect is not considered here.

The task of candidate star identification is done using the normalized correlation described in Section 4.2.1. Recall that normalized correlation is especially useful for template matching independent of image brightness. This allows for the same template to be used to find stars of varying magnitude. Because a bivariate Gaussian PSF is known to be a good approximate of a blurred point source, the template used to find candidate stars is a Gaussian kernel (see Fig 4.7.a). Responses in the filtered image above a specified threshold are taken to be candidate star locations. Next, a connected component analysis is done so that adjacent pixels that are above the threshold are taken to belong to the same star.

Once initial estimates of the candidate star locations are known, a box is drawn around each location. Then the average intensity associated with the detector dark current (and other background noise) is subtracted off of each pixel. Finally, the weighted average approach described in Section 3.10.1 is used to compute the candidate star centroid location.

As an example, this technique is applied to an image taken by one of

the Clementine star trackers. This image contains both stars and a portion of the limb of the Moon. As is clear from Fig. 4.16, this approach identifies both stars and some surface features that happen to have the same two-dimensional light intensity pattern as a blurred star. This technique may also incorrectly identify other non-star objects such as asteroids, other spacecraft, planets, moons, or any other object with a two-dimensional light intensity pattern that matches the bivariate Gaussian template.

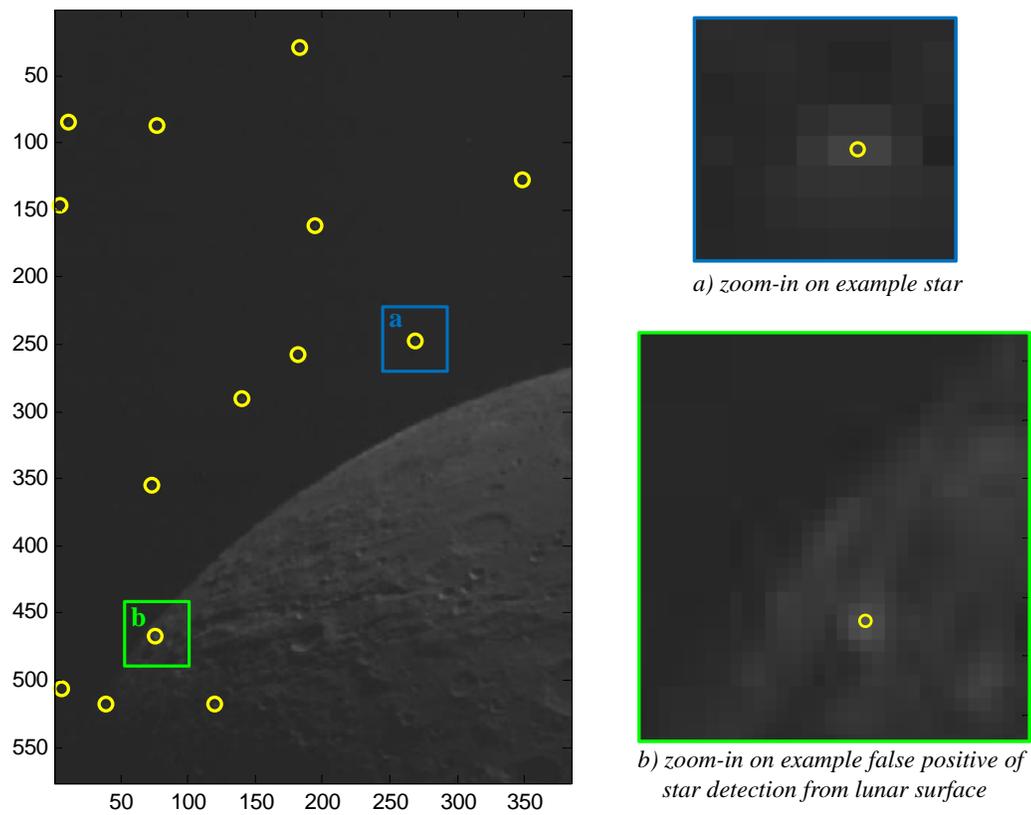


Figure 4.16: Example result of star finding algorithm on a raw image taken by the Clementine spacecraft’s Star Tracker B on 20 March 2004. Raw image is Product ID from LBA0032V.137 from [94].

There are numerous ways to determine which candidate stars are real stars, and which ones are false positives. The first method is to use an *a-priori* attitude estimate to predict the location of stars in the image. Reference star directions in the inertial frame may be determined from a number of different star catalogs, as discussed in Appendix B. Candidate stars that appear near to expected star locations are assumed to be the predicted star. If no *a-priori* attitude estimate is available, there are numerous star pattern recognition algorithms that may be used to identify real stars and discard the non-star objects. These algorithms are used regularly on commercially available star trackers and are well understood. An algorithm of particular note, and the one used in subsequent analyses to solve this problem, is the Pyramid Star Identification algorithm developed by Mortari. The details of this algorithm are provided in [144] and [145].

4.7 New Algorithm for Planet Detection and Optical Navigation Measurements

The autonomous generation of navigation measurements from a raw image is a multistep process. The first step is to rotate the spacecraft or the image so that the Sun appears to enter from the left side of the image. The second step is to perform autonomous planet/moon finding in the rotated image. The third step is to extract useful navigation measurements from the image. For this third step, two different measurement types will be discussed: (1) the use of planet centroid and apparent diameter to generate an estimate

of spacecraft position relative to the planet/moon and (2) the angle between the planet/moon horizon and a reference star.

The details of these three steps are provided in the following subsections.

4.7.1 Step 1: Rotate for Proper Sun Orientation

The first step is to control the approximate direction of illumination in the image. Because planets are assumed to be primarily illuminated by sunlight, controlling the direction from which sunlight enters the image can significantly simplify the subsequent steps. Although the choice of this direction is arbitrary, it was chosen for objects to be illuminated from the left side of the image (i.e. sunlight enters from the left side of the image and moves in the positive u direction). If the Sun is assumed to be far away from the planet/moon and spacecraft, then this is achieved by making the projection of the Sun line-of-sight vector in the image plane lie along the positive u -axis. The positive u -axis is required because the raw image appears flipped with respect to the actual geometry (see Fig 4.17).

The primary difficulty here lies with determining the direction to the Sun. This may be achieved directly by an instrument such as a Sun sensor. Alternatively, if a good *a-priori* state estimate is available, the direction to the Sun may be computed using the onboard estimate of the spacecraft position and attitude. If neither of these measurements are available, it may be possible to determine the light source direction from information available in

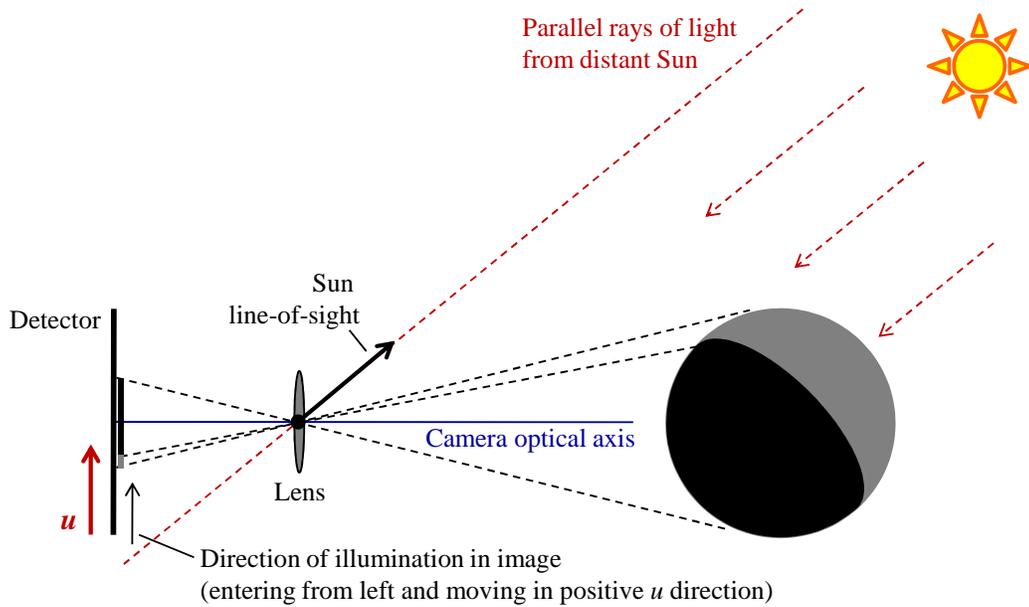


Figure 4.17: Illumination direction.

the image. Techniques for performing this task are discussed in [146], [147], and [131]. These methods use the light intensity gradient on the surface of the observed object to obtain the direction to the light source. To use these methods, one would begin with Step 2 to determine the pixels corresponding to the illuminated planet/moon surface. Then these algorithms would only be applied to the illuminated pixels. With the illumination direction known, the image may be rotated into the proper orientation. The remainder of this discussion, however, will assume that an approximate line-of-sight to the Sun is available from one of the first two methods.

To align the projection of the Sun line-of-sight vector with the u -axis requires either (1) the camera to be physically rotated to the proper orientation

or (2) an image preprocessing algorithm that rotates an arbitrary image to the proper orientation. The first method is the simplest from the perspective of the planet detection algorithm because it requires no rotational preprocessing of the image. Other constraints on the spacecraft (e.g. thermal, power, science, etc.), however, may make it undesirable to orient the navigation camera as described above. If the unit vector to the Sun is known, then the required image rotation (which is equivalent to a physical camera rotation about the camera optical axis) is also known.

When the original image is rotated into a new image, the rotated coordinates of the old pixel locations will not line up with the pixel locations of the new image (see Fig. 4.18). Therefore, a bilinear interpolation routine is used to determine the light intensity at each pixel location in the new image. This is a standard task in image processing and may be performed using the “imrotate” command in MATLAB’s Image Processing Toolbox.

It is also necessary to place a restriction on the minimum allowable

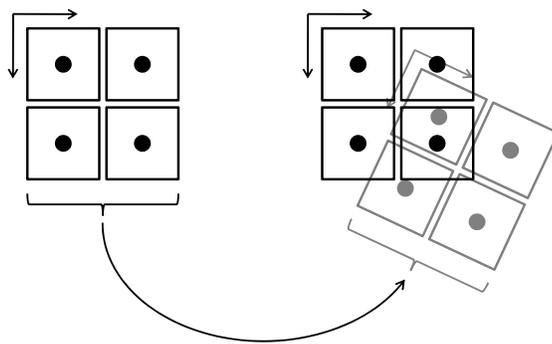


Figure 4.18: Image rotation geometry.

angle between the camera optical axis and the line-of-sight to the Sun (typically called the Sun exclusion angle). This restriction serves a number of purposes. First, scattering of light from the Sun (even if it is occluded by a planet or moon in the image) can cause difficulties with accurately determining the limb location. It also causes problems with detecting faint stars that may be required for navigation. Secondly, some types of detectors may be damaged by directly imaging the Sun with the exposure durations used to generate OPNAV images.

4.7.2 Step 2: Autonomous Planet/Moon Finding and Registration

The second step is to find planets or moons in the image. Begin this process by thresholding the image. An estimate of the appropriate threshold setting may be found from the optical model described in Section 3.9.2 along with knowledge of the detector well depth. Next, close the image as discussed in Section 4.2.3 to connect regions of the same planet/moon that may have been separated in the thresholding step by small shadows. A circular structuring element is used here. Now perform a connected components analysis on the closed binary image and keep the largest group that is above a specified number of pixels in size. If there are no groups that meet the minimum pixel size requirement, then it is assumed that there is no planet/moon in the image. If more than one planet/moon is visible, and more than one connected components group is used, it may be possible to derive measurements from multiple celestial bodies using a single image. Using more than one group to

detect multiple bodies requires additional logic checks that are not discussed here.

Looking only at the largest group from the connected components analysis, find the first foreground pixel in each row of the image. These points are the left-side edge points for this group. Because of the rotation performed in Step 1, most of these points will correspond to the lit limb of the planet or moon. Some of the points returned by this process will not be limb points due to features such as cratering, variations in surface albedo, detector noise, and other factors. The MSAC algorithm will be used to proceed with performing planet/moon registration in the presence of these errors.

Finally, an ellipse is fit to the candidate horizon points using the improved ellipse fitting technique described in Section 4.4.2. Because the set of candidate horizon points may contain a substantial number of outliers that were not removed in the previous processing steps, the MSAC algorithm is used to fit the elliptical model to the data. The geometric distance, rather than the algebraic distance, is used as the distance criteria in Eq. 4.77. This choice was made because the geometric distance has a consistent physical interpretation regardless of the point's location on the ellipse. Algebraic distance increases at a slower rate as the radius of curvature decreases, while geometric distance increases at the same rate everywhere (regardless of the radius of curvature).

This procedure provides an extremely robust method for finding a planet/moon in an OPNAV image. With very little modification, this methodology can be applied to a wide range of problems. Consider the different sce-

narios shown in Fig. 4.19 through Fig. 4.24. All of the images used in these examples are actual images taken aboard the spacecraft indicated in the figure captions. The procedure described above was applied to all of these cases with no modification. The only parameter that was changed from example to example was the threshold, τ . The value of τ used in each example is also given in the caption. It is clear from these images that this algorithm may be applied to numerous scenarios with excellent performance.

Looking at Fig. 4.21 and Fig. 4.22, it is also interesting to note that the solution is not extremely sensitive to the choice of τ . The biggest difficulty with using a threshold that is too high is that regions of shadow (or comparatively low albedo) may separate the lit regions of the planet. The combination of closing the image and using MSAC to fit the horizon ellipse make this algorithm relatively robust to this problem.

The case of Fig. 4.24 allows for an important observation regarding this algorithm. Initial tests indicate that the algorithm will not reliably identify the centroid of a planet/moon unless more than about 1/4 of the limb is used to fit the ellipse. When only a small portion of the horizon is used to fit the ellipse, the algorithm usually provides a good local fit (as seen in Fig. 4.24), but is unable to accurately find the centroid. Fortunately, as will be discussed in the next section, it is easy to determine if this erroneous case occurs. In such a scenario, the centroid and apparent diameter measurement is not made, and only the star-horizon measurements are available.

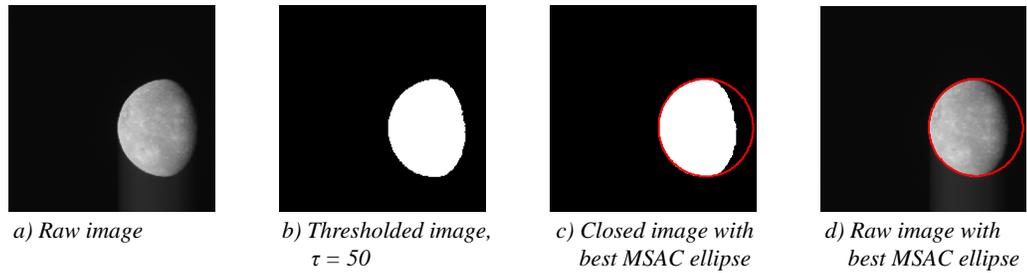


Figure 4.19: Planet finding algorithm applied to example image containing Mercury taken by MESSENGER spacecraft's Narrow Angle Camera on 15 January 2008. Raw image is Product ID EN0108892844M from [93].

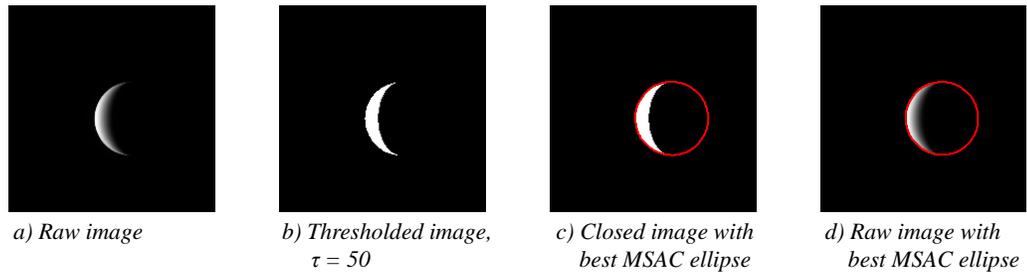


Figure 4.20: Planet finding algorithm applied to example image containing Venus taken by MESSENGER spacecraft's Narrow Angle Camera on 6 June 2007. Raw image is Product ID CN0089716371M_IF_3 from [67].

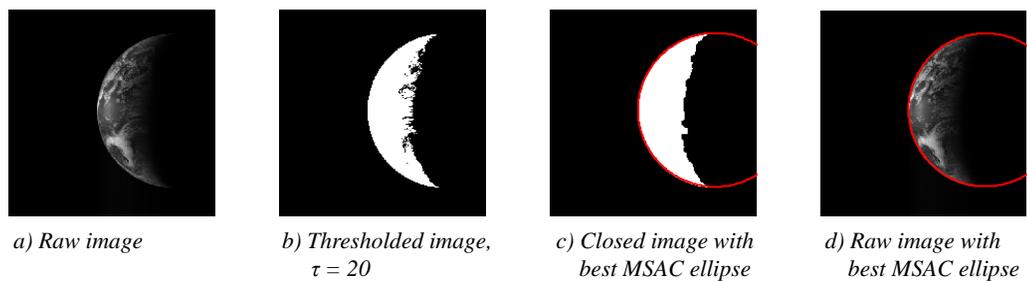


Figure 4.21: Planet finding algorithm applied to example image containing Earth taken by MESSENGER spacecraft's Narrow Angle Camera on 2 August 2005. Raw image is Product ID EW0031513371D from [93].

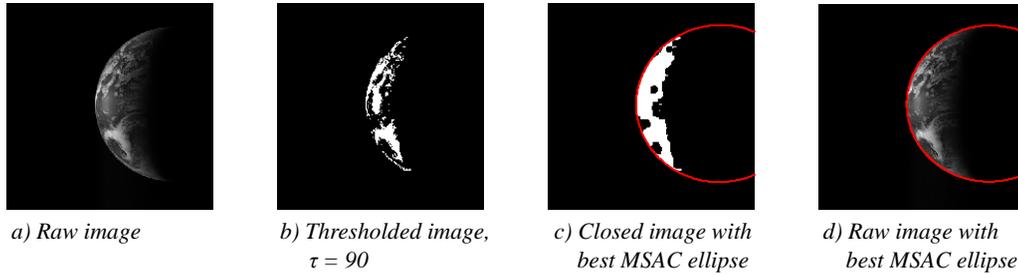


Figure 4.22: Planet finding algorithm applied to example image containing Earth taken by MESSENGER spacecraft's Narrow Angle Camera on 2 August 2005. Raw image is Product ID EW0031513371D from [93].

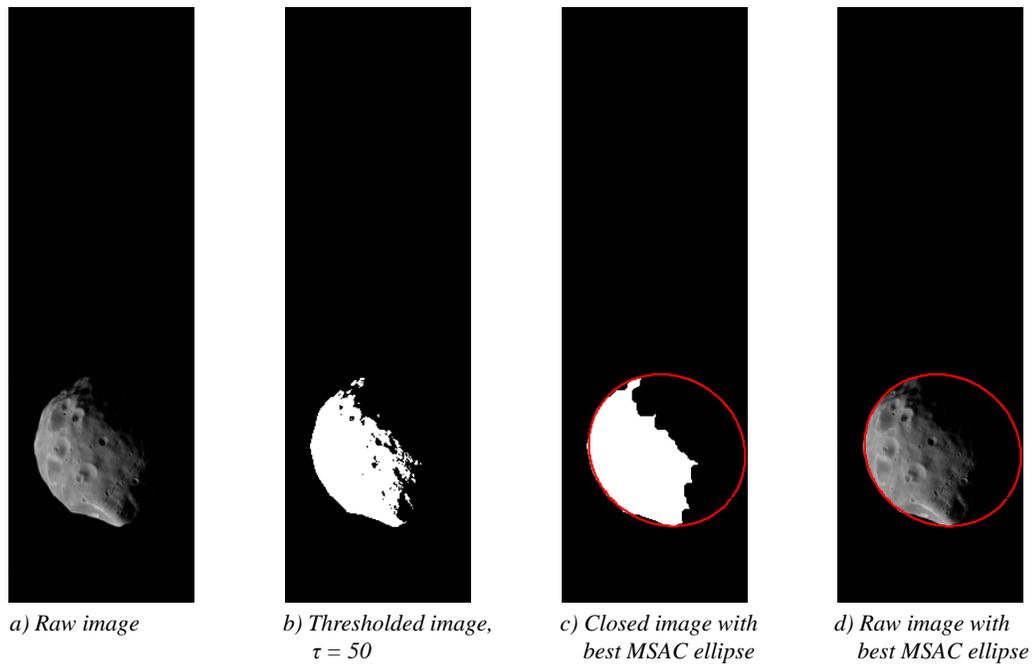


Figure 4.23: Planet finding algorithm applied to example image containing Phobos taken on 22 August 2004 by the Mars Orbiter Camera (MOC) on ESA's Mars Express spacecraft. Raw image is Product ID H0756_0000_P12 from [148].

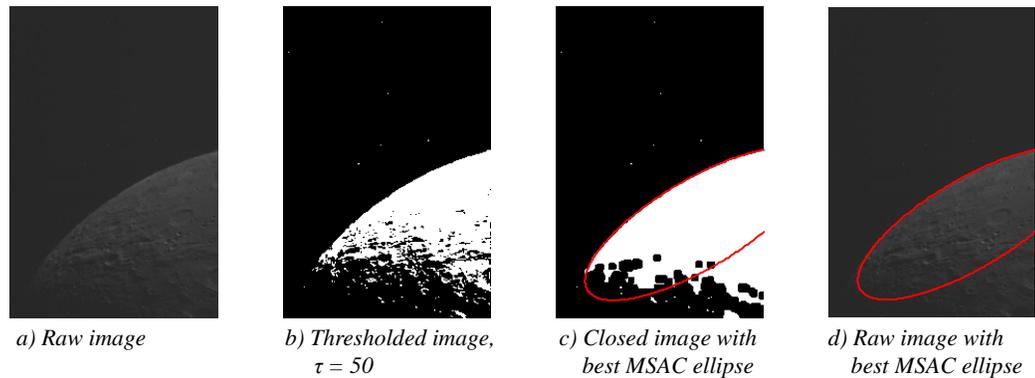


Figure 4.24: Planet finding algorithm applied to example image containing lunar horizon taken by the Clementine spacecraft’s Star Tracker B on 20 March 2004. Raw image is Product ID from LBA0032V.137 from [94].

4.7.3 Step 3: Extraction of Navigation Measurements

4.7.3.1 Spacecraft Position from Centroid and Apparent Diameter

In most cases, the results of the planet finding procedure in Step 2 will provide enough information to directly compute an estimate of the spacecraft location. The ellipse parameters include the location of the ellipse centroid and the dimensions/orientation of the ellipse in the image plane. Geometry will show, however, that the line-of-sight vector produced by the coordinates of the ellipse center, $[x_0, y_0]$, does not point towards the planet center. The actual line-of-sight vector to the planet center was computed through a simple iterative process that typically converges in 3-4 iterations. This routine, which was developed by the author, is discussed in Appendix C.

Therefore, if the line-of-sight unit vector to the planet center is known,

then the camera position vector with respect to the planet is given by

$$\mathbf{r}_c = -\rho(\mathbf{e}_c)_I \quad (4.78)$$

where ρ is the range from the camera to the planet center and $(\mathbf{e}_c)_I$ is the line-of-sight unit vector from the camera to the planet center expressed in the inertial frame. Therefore, if the range ρ is known, then the spacecraft position will also be known.

To estimate ρ , select a point on the ellipse that was fit to the horizon segment in Step 2. Each point on this ellipse is assumed to lie on the horizon and will produce a line-of-sight vector in the inertial frame denoted by $(\mathbf{e}_i)_I$. Recalling the relation from Eq. 3.86, any observed point on the planet \mathbf{p}_i is given by

$$\mathbf{p}_i = \mathbf{s}_i + \mathbf{r}_c = t(\mathbf{e}_i)_I - \rho(\mathbf{e}_c)_I \quad (4.79)$$

If the observed point is on the horizon, the line-of-sight vector \mathbf{s}_i must be perpendicular to the surface normal, \mathbf{p}_n . Recalling the surface normal relation for an ellipse from Eq. 3.99,

$$\mathbf{s}_i^T \mathbf{p}_n = (\mathbf{p}_i - \mathbf{r}_c)^T \mathbf{A} \mathbf{p}_i = 0 \quad (4.80)$$

and because \mathbf{p}_i is constrained to lie on the ellipse,

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_i = 1 \quad (4.81)$$

it is clear that the following is also true

$$\mathbf{r}_c^T \mathbf{A} \mathbf{p}_i = 1 \quad (4.82)$$

Substituting for \mathbf{p}_i into Eq. 4.81 and Eq. 4.82 yields the following two equations for ρ and t that must be solved simultaneously

$$\rho^2 \left[(\mathbf{e}_c)_I^T \mathbf{A} (\mathbf{e}_c)_I \right] - \rho t \left[(\mathbf{e}_c)_I^T \mathbf{A} (\mathbf{e}_i)_I \right] = 1 \quad (4.83)$$

$$\rho^2 \left[(\mathbf{e}_c)_I^T \mathbf{A} (\mathbf{e}_c)_I \right] - 2\rho t \left[(\mathbf{e}_c)_I^T \mathbf{A} (\mathbf{e}_i)_I \right] + t^2 \left[(\mathbf{e}_i)_I^T \mathbf{A} (\mathbf{e}_i)_I \right] = 1 \quad (4.84)$$

Solving these two equations yields the following expression for ρ

$$\rho = \left\{ \left[(\mathbf{e}_c)_I^T \mathbf{A} (\mathbf{e}_c)_I \right] - \frac{\left[(\mathbf{e}_c)_I^T \mathbf{A} (\mathbf{e}_i)_I \right]^2}{\left[(\mathbf{e}_i)_I^T \mathbf{A} (\mathbf{e}_i)_I \right]} \right\}^{-\frac{1}{2}} \quad (4.85)$$

If the range is known, the position of the spacecraft relative to the observed planet is known from Eq. 4.78.

This range computation may be used in two different ways to generate the measured range. The first method is to use only one horizon point to create the range measurement. If the image has been rotated as described in Fig. 4.17, it was found that the best horizon point to use is the one that lies along the negative u -direction from the centroid of the ellipse generated in Step 2 (approximately the center of the lit horizon).

The second approach is to perform this range computation at a number of horizon locations on the ellipse generated in Step 2. The average range from this sweep is taken as the measured range. A preliminary analysis showed that this procedure did not offer a substantial improvement in the measured range over the single horizon point technique. This method, however, does allow for

the computation of the standard deviation of the set of range measurements generated by this sweep:

$$\tilde{\sigma}_\rho = \sqrt{\frac{1}{n} \sum_{i=1}^n (\rho_i - \mu_\rho)^2} \quad \mu_\rho = \frac{1}{n} \sum_{i=1}^n \rho_i \quad (4.86)$$

where ρ_i is the i -th computed range, n is the number of horizon points used, μ_ρ is the mean of the set of computed ranges, and $\tilde{\sigma}_\rho$ is the numerically computed standard deviation of the set of computed ranges (not to be confused with the standard deviation of the range measurement). Therefore, $\tilde{\sigma}_\rho$ acts as a measure of the consistency of the ranges generated by the ellipse as the reference point used to compute the range is swept around the ellipse. If the standard deviation is small, then the ellipse is likely a good fit. If the standard deviation is too large, then the ellipse generated in Step 2 is not a good fit and any range generated from Eq. 4.85 may not be valid. The incorrect ellipse generated in Fig. 4.24, for example, would fail this standard deviation test.

Further refinement of the measured spacecraft position is possible. The measured state generated using the ellipse from Step 2 may be used to create an expected planet/moon lighting profile. The information contained in this two-dimensional lighting profile, generated using the models from Section 3.9.2, may be compared to the actual lighting profile seen in the image.

Initial attempts to match the expected planet/moon lighting profile to the image data employed an iterative batch estimation approach in the two-dimensional region surrounding the planet/moon in the image. The partial derivatives of the light intensity with respect to the estimated parameters

were used to perform the updates at each iteration. Performing this type of procedure on the lighting profile of the planet/moon as a whole proved to be unstable (or generate undesirable biases) for many lighting configurations. This batch estimation approach was attempted with two different sets of parameters being estimated: (1) direct estimation of the spacecraft position and (2) estimation of the planet/moon centroid in the image. The root of the difficulty is thought to lie with the light intensity gradient becoming undefined for rays that do not intersect the planet/moon. Therefore, if the update shifts a point on the image plane to a location whose line-of-sight vector no longer intersects the planet/moon, there is no way to determine the direction or magnitude of the next update. Therefore, a horizon scan approach was chosen to address this problem.

Horizon scan directions are chosen at regular intervals spanning the lit horizon of the planet/moon. Next, the expected horizon location in that scan direction is found. A strip of data centered around this horizon location is then extracted from the image along the scan direction. Data points along this strip are created at intervals of 1/10 of a pixel using bilinear interpolation.

The expected light intensity at each location along the strip is also computed using the procedure described in Section 3.9.2. This strip is then moved up and down the scan direction to find the location where the residuals between the two one-dimensional lighting profiles are minimized. The point with the minimum residual is then taken as the updated horizon location. After all the horizon points have been updated, a new ellipse is fit to the

data to create an updated estimate of the center location. The position of the camera relative to the observed planet/moon is then recomputed using Eq. 4.78 and Eq. 4.85.

The only remaining mathematical difficulty, therefore, is finding the expected horizon location in the image given an estimate of the camera position relative to the target planet/moon. Recalling the results from Eq. 3.89, the points on the observed horizon are constrained by

$$\left[(\mathbf{e}_i)_I^T \mathbf{A} \mathbf{r}_c + \mathbf{r}_c^T \mathbf{A} (\mathbf{e}_i)_I \right]^2 - 4 \left[(\mathbf{e}_i)_I^T \mathbf{A} (\mathbf{e}_i)_I \right] \left[\mathbf{r}_c^T \mathbf{A} \mathbf{r}_c - 1 \right] = 0 \quad (4.87)$$

This may be factored and rewritten in terms of the line-of-sight vector in the camera frame as

$$(\mathbf{e}_i)_C^T \mathbf{T}_C^I \left[\mathbf{A} \mathbf{r}_c^T \mathbf{r}_c \mathbf{A} - (\mathbf{r}_c^T \mathbf{A} \mathbf{r}_c - 1) \mathbf{A} \right] \mathbf{T}_I^C (\mathbf{e}_i)_C = 0 \quad (4.88)$$

If Eq. 3.22 is substituted for $(\mathbf{e}_i)_C$ and the symmetric matrix \mathbf{M} is defined as

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{12} & M_{22} & M_{23} \\ M_{13} & M_{23} & M_{33} \end{bmatrix} = \mathbf{T}_C^I \left[\mathbf{A} \mathbf{r}_c^T \mathbf{r}_c \mathbf{A} - (\mathbf{r}_c^T \mathbf{A} \mathbf{r}_c - 1) \mathbf{A} \right] \mathbf{T}_I^C \quad (4.89)$$

then Eq. 4.88 may be rewritten as

$$\begin{bmatrix} -x_i & -y_i & Z_l \end{bmatrix} \mathbf{M} \begin{bmatrix} -x_i \\ -y_i \\ Z_l \end{bmatrix} = 0 \quad (4.90)$$

Expanding this matrix equation, allows for this to be rewritten in the standard implicit form for a conic section

$$F(x_i, y_i) = Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F = 0 \quad (4.91)$$

with

$$A = M_{11} \quad B = 2M_{12} \quad C = M_{22} \quad (4.92)$$

$$D = -2Z_l M_{13} \quad E = -2Z_l M_{23} \quad F = Z_l^2 M_{33} \quad (4.93)$$

The standard ellipse parameters may be found from the implicit equation coefficients using Eq. 4.16 through Eq. 4.19. Therefore, if a scan direction is known, then the expected horizon location is also known. This allows for the creation of the one dimensional lighting profiles necessary for the improved horizon location measurement. This result also demonstrates that the perspective projection of an ellipsoid onto the image plane is an ellipse.

The additional step of minimizing the lighting profile residuals typically provides a significant improvement in performance. Most situations (on both real and synthetic images) show a reduction in position error by a factor of 2 to 5, depending on lighting and geometry. As an example, consider a $10.5^\circ \times 10.5^\circ$ FOV camera with a 1024×1024 pixel detector that images Earth's Moon from a range of 22,433 km. A Monte Carlo analysis was run on this situation and synthetic images of the Moon were generated. The results, shown in 4.25, are representative of the improvement that may be obtained by minimizing the lighting profile residual.

4.7.3.2 Angle Between Horizon and Star

The second type of optical navigation measurement is the angle between the planet/moon horizon and a reference star. As was mentioned earlier, this

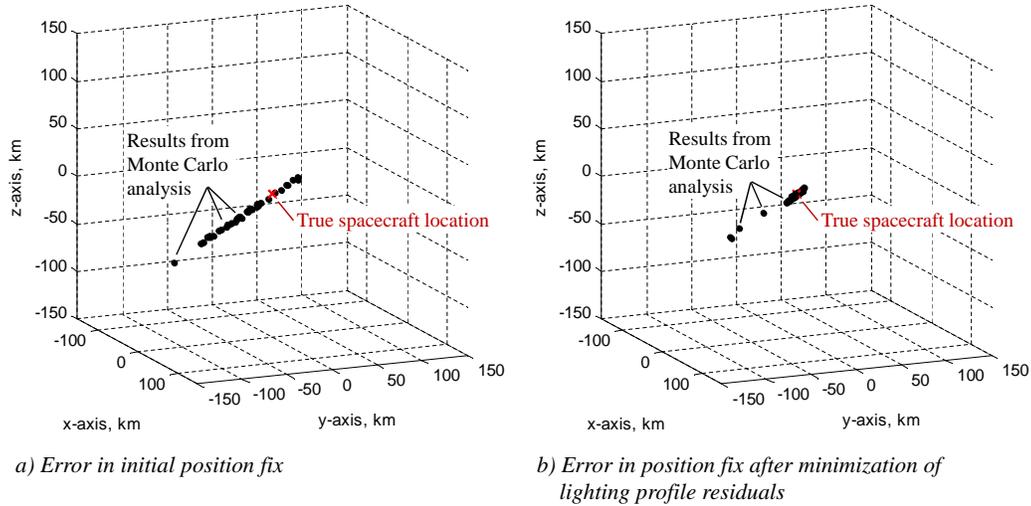


Figure 4.25: Error ellipse for instantaneous position fix from centroid and apparent diameter, with and without minimization of light profile residuals. Results for simulated image of Moon at a range of 22,433 km with a $10.5^\circ \times 10.5^\circ$ FOV camera.

type of measurement is still possible even if the image processing algorithm only provides a good elliptical fit to a portion of the horizon.

Suppose that the measured line-of-sight vector to a star (which may be identified using one of the methods discussed in Section 4.6) is given by the unit vector $\tilde{\mathbf{e}}_s$. Next, given an ellipse, the point on that ellipse closest to the star may be found by from Eq. 4.36 and Eq. 4.26. If the ellipse only provides a good fit to a portion of the horizon, then only ellipse coordinates that lie close to the measured horizon are permitted. Let the line-of-sight vector to this horizon point be given by the unit vector $\tilde{\mathbf{e}}_h$. Therefore, the measured

angle between the horizon and reference star, $\tilde{\psi}$, is given by

$$\tilde{\psi} = \text{acos} [\tilde{\mathbf{e}}_s^T \tilde{\mathbf{e}}_h] \quad (4.94)$$

Although it does not matter what frame these vectors are expressed in (any frame will give the same angle), it is easiest to keep everything in the camera frame.

As an example, consider the Clementine star tracker image used in a number of previous examples. Specifically, suppose that the results of the star finding step shown in Fig. 4.16 are combined with the results of the horizon finding algorithm shown in Fig. 4.24. Numerous star-horizon measurements may now be made directly as shown in Fig. 4.26. Although all possible star-horizon measurements are shown in this figure, not all of these would necessarily be included into a navigation filter.

4.8 Comments on Image Exposure Time

The selection of the image exposure time is extremely important. If exposure time is too short, the detector will not have time to collect a large number of photons from the source and the resulting image will have a low signal-to-noise ratio and will be too dark. If the exposure time is too long, pixel saturation may become a problem. That being said, pixel saturation may be unavoidable in some scenarios. Further, some algorithms have proposed intentionally overexposing the image for certain applications. Of particular note in the present context is a recent algorithm that uses overexposed images

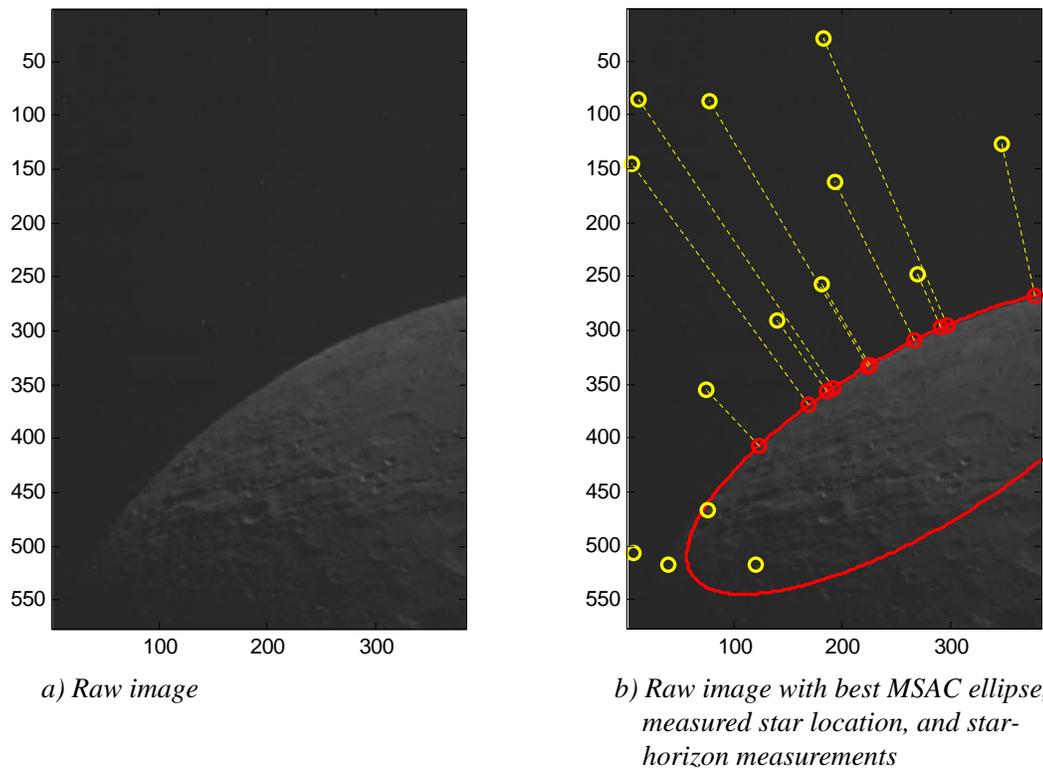


Figure 4.26: Example star-horizon measurements from image taken by the Clementine Star Tracker B on 20 March 2004. Raw image is Product ID from LBA0032V.137 from [94].

of the Moon to compute attitude.[149]

The most apparent problem with long exposure times and overexposed images is that light intensity gradient information is lost as pixels become saturated. This can cause numerous difficulties in an optical navigation algorithm. Recall, for example, that the centroid and apparent diameter measurement had a refinement step that minimized the residuals between a predicted lighting profile and the observed lighting profile. If the image is overexposed, then

the observed lighting profile is nothing more than a constant intensity value (at the maximum) for a large portion of the pixels of interest. For many types of lighting configurations, the brightest part of the planet is the horizon (a direct result of the Lommel-Seeliger Law) - which is precisely the place that we do not want to lose information. It is worth pointing out, however, that losing gradient information may not always be bad. Overexposing an image does remove local features and shadowing from the image and may largely eliminate the need for the dilation and erosion step in the image processing algorithm.

The second problem with overexposing an image is that stray light from dust surrounding the planet and the spacecraft (and light scattered by the planet's atmosphere, if it has one) will accumulate and may cause the planet to appear larger than it actually is. Because the pixels are saturated, there is no way to reverse this effect in post processing. The end result of such an effect is the introduction of a bias in the range measurement, making the planet appear closer (larger) than it should. It may be possible to estimate this bias in some applications as an unknown increase in the planet radius.

The third problem, if the detector is a CCD, is that overexposure will introduce blooming. As the CCD pixels become saturated, charge will begin to spill over into adjacent pixels in the same column (or row). The problem of blooming, however, can be largely eliminated by selecting a CMOS detector instead.

Other problems come simply from having a longer exposure time. In

addition to allowing more time to accumulate photons, long exposure times also allow more time to integrate noise sources, such as dark current. Long exposure times can also lead to smearing in the image. If the vehicle undergoes significant rotation during the exposure, stars and other objects will appear smeared in the image. Although image processing algorithms capable of removing the bulk of this effect exist, this creates an additional source of error.

As was mentioned earlier, it may be impossible to avoid overexposure in some cases. For spacecraft performing operations near one of the inner planets, it is difficult to capture stars and the planet in the same image without overexposing the planet. Therefore, the star-horizon measurement will almost always be made with an overexposed planet.

Fortunately, the response of most CCD and CMOS detectors varies in a nearly linear fashion with integration time when viewing a source of constant intensity. Therefore, doubling the integration time will double the amount of photoelectrons generated. It is worth noting that this linear relationship does break down near the pixel saturation point (especially in CCDs with anti-blooming drains), which is another drawback of imaging near (or above) the pixel saturation point. Pixels on a modern CCD can maintain their linear response up to about 90% of their saturation point.[107] Therefore, it is recommended that exposure times be selected to avoid nearly saturating detector pixels.

4.9 Optical Navigation Measurement Covariance

4.9.1 Spacecraft Position from Centroid and Apparent Diameter

The position of the camera relative to the observed planet or moon may be estimated from the centroid and apparent diameter measurements. The covariance of this camera position measurement is defined as

$$\mathbf{R}_{\mathbf{r}_c} = E \left[(\tilde{\mathbf{r}}_c - \mathbf{r}_c) (\tilde{\mathbf{r}}_c - \mathbf{r}_c)^T \right] \quad (4.95)$$

From Eq. 4.78, the expression for \mathbf{r}_c is known to be

$$\mathbf{r}_c = -\rho (\mathbf{e}_c)_I = -\rho \mathbf{T}_I^C (\mathbf{e}_c)_C \quad (4.96)$$

$$\tilde{\mathbf{r}}_c = -\tilde{\rho} \mathbf{T}_I^{\hat{C}} (\tilde{\mathbf{e}}_c)_C \quad (4.97)$$

Error is introduced into the estimate of \mathbf{r}_c from three sources: (1) error in the measured range, $\delta\rho$, (2) error in the measured line-of-sight unit vector expressed in the camera frame, $\boldsymbol{\epsilon}_c$, and (3) error in the current estimate of the spacecraft attitude, $\boldsymbol{\delta\theta}$. Define these three errors as

$$\delta\rho = \tilde{\rho} - \rho \quad (4.98)$$

$$\boldsymbol{\epsilon}_i = (\tilde{\mathbf{e}}_i)_C - (\mathbf{e}_i)_C \quad \boldsymbol{\epsilon}_c = (\tilde{\mathbf{e}}_c)_C - (\mathbf{e}_c)_C \quad (4.99)$$

$$\mathbf{T}_C^{\hat{C}} = \mathbf{I}_{3 \times 3} - [\boldsymbol{\delta\theta} \times] \quad (4.100)$$

Substituting these into Eq. 4.97 yields the following expression for the estimated \mathbf{r}_c

$$\tilde{\mathbf{r}}_c = -(\rho + \delta\rho) \mathbf{T}_I^C [\mathbf{I}_{3 \times 3} - [\boldsymbol{\delta\theta} \times]] [(\mathbf{e}_c)_C + \boldsymbol{\epsilon}_c] \quad (4.101)$$

Therefore, the term $(\tilde{\mathbf{r}}_c - \mathbf{r}_c)$ in Eq. 4.95 may be expanded as

$$\begin{aligned} \tilde{\mathbf{r}}_c - \mathbf{r}_c = & -\rho \mathbf{T}_I^C \boldsymbol{\epsilon}_c - \delta\rho \mathbf{T}_I^C (\mathbf{e}_c)_C - \delta\rho \mathbf{T}_I^C \boldsymbol{\epsilon}_c - \rho \mathbf{T}_I^C [(\mathbf{e}_c)_C \times] \boldsymbol{\delta\theta} \\ & - \rho \mathbf{T}_I^C [\boldsymbol{\epsilon}_c \times] \boldsymbol{\delta\theta} - \delta\rho \mathbf{T}_I^C [(\mathbf{e}_c)_C \times] \boldsymbol{\delta\theta} - \delta\rho \mathbf{T}_I^C [\boldsymbol{\epsilon}_c \times] \boldsymbol{\delta\theta} \end{aligned} \quad (4.102)$$

Recall from Eq. 4.85 that ρ is a function of the line-of-sight to the planet center, $(\mathbf{e}_c)_I$, and the line of sight to a specified horizon point, $(\mathbf{e}_i)_I$. Therefore, taking the Taylor series expansion of the range and keeping only first order terms will yield,

$$\delta\rho = \frac{\partial\rho}{\partial(\mathbf{e}_c)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_c + \frac{\partial\rho}{\partial(\mathbf{e}_i)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_i \quad (4.103)$$

The partial derivatives may be easily computed by the direct differentiation of Eq. 4.85 and are given by:

$$\begin{aligned} \frac{\partial\rho}{\partial(\mathbf{e}_i)_I} = \frac{1}{\rho^3} \left\{ \frac{[(\mathbf{e}_c)_I^T \mathbf{A}(\mathbf{e}_i)_I]}{[(\mathbf{e}_i)_I^T \mathbf{A}(\mathbf{e}_i)_I]} (\mathbf{e}_c)_I^T \mathbf{A} \right. \\ \left. - \left(\frac{[(\mathbf{e}_c)_I^T \mathbf{A}(\mathbf{e}_i)_I]}{[(\mathbf{e}_i)_I^T \mathbf{A}(\mathbf{e}_i)_I]} \right)^2 (\mathbf{e}_i)_I^T \mathbf{A} \right\} \end{aligned} \quad (4.104)$$

$$\frac{\partial\rho}{\partial(\mathbf{e}_c)_I} = \frac{1}{\rho^3} \left\{ -(\mathbf{e}_c)_I^T \mathbf{A} + \frac{[(\mathbf{e}_c)_I^T \mathbf{A}(\mathbf{e}_i)_I]}{[(\mathbf{e}_i)_I^T \mathbf{A}(\mathbf{e}_i)_I]} (\mathbf{e}_i)_I^T \mathbf{A} \right\} \quad (4.105)$$

Therefore, substitute Eq. 4.103 into Eq. 4.102 and combining like terms,

$$\begin{aligned}
\mathbf{r}_c - \tilde{\mathbf{r}}_c = & \left[\rho + \mathbf{T}_I^C (\mathbf{e}_c)_C \frac{\partial \rho}{\partial (\mathbf{e}_c)_I} + \frac{\partial \rho}{\partial (\mathbf{e}_c)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_c + \frac{\partial \rho}{\partial (\mathbf{e}_i)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_i \right] \mathbf{T}_I^C \boldsymbol{\epsilon}_c \quad (4.106) \\
& + \mathbf{T}_I^C (\mathbf{e}_c)_C \frac{\partial \rho}{\partial (\mathbf{e}_i)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_i \\
& + \left[\rho + \frac{\partial \rho}{\partial (\mathbf{e}_c)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_c + \frac{\partial \rho}{\partial (\mathbf{e}_i)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_i \right] \mathbf{T}_I^C [(\mathbf{e}_c)_C \times] \boldsymbol{\delta \theta} \\
& + \left[\rho + \frac{\partial \rho}{\partial (\mathbf{e}_c)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_c + \frac{\partial \rho}{\partial (\mathbf{e}_i)_I} \mathbf{T}_I^C \boldsymbol{\epsilon}_i \right] \mathbf{T}_I^C [\boldsymbol{\epsilon}_c \times] \boldsymbol{\delta \theta}
\end{aligned}$$

Now, proceed by assuming that $\boldsymbol{\epsilon}_i$, $\boldsymbol{\epsilon}_c$, and $\boldsymbol{\delta \theta}$ are uncorrelated and that higher order moments (skewness, kurtosis, etc.) are unimportant. Therefore, the covariance is given by

$$\begin{aligned}
\mathbf{R}_{\mathbf{r}_c} = & \left[\rho + \mathbf{T}_I^C (\mathbf{e}_c)_C \frac{\partial \rho}{\partial (\mathbf{e}_c)_I} \right] \mathbf{R}_{I,c} \left[\rho + \mathbf{T}_I^C (\mathbf{e}_c)_C \frac{\partial \rho}{\partial (\mathbf{e}_c)_I} \right]^T \quad (4.107) \\
& + \left[\mathbf{T}_I^C (\mathbf{e}_c)_C \frac{\partial \rho}{\partial (\mathbf{e}_i)_I} \right] \mathbf{R}_{I,i} \left[\mathbf{T}_I^C (\mathbf{e}_c)_C \frac{\partial \rho}{\partial (\mathbf{e}_i)_I} \right]^T \\
& + \rho^2 \mathbf{T}_I^C [(\mathbf{e}_c)_C \times] \mathbf{P}_{\theta\theta} [(\mathbf{e}_c)_C \times]^T \mathbf{T}_I^C
\end{aligned}$$

where

$$\mathbf{P}_{\theta\theta} = E [\boldsymbol{\delta \theta} \boldsymbol{\delta \theta}^T] \quad (4.108)$$

$$\mathbf{R}_{I,c} = \mathbf{T}_I^C \mathbf{R}_{C,c} \mathbf{T}_I^I \quad \mathbf{R}_{I,i} = \mathbf{T}_I^C \mathbf{R}_{C,i} \mathbf{T}_I^I \quad (4.109)$$

if $\mathbf{R}_{C,c}$ and $\mathbf{R}_{C,i}$ are from Eq. 3.158. Therefore, the result given in Eq. 4.107 provides an analytic expression for the covariance of the instantaneous position fix measurement generated from the centroid and apparent diameter extracted from an optical navigation image.

As an example, consider a $1.5^\circ \times 1.5^\circ$ FOV camera with a 1024×1024 pixel detector. Suppose this camera images Earth's Moon from a range of 200,636 km. The instantaneous position fix covariance from a centroid and apparent diameter measurement associated with this scenario is shown in Fig. 4.27. To verify this covariance model, a Monte Carlo analysis was performed on synthetically generated images of the Moon. The results of this Monte Carlo analysis were superimposed on the analytic covariance matrix.

The large axis of the covariance matrix in Fig. 4.27 is along the line-of-sight direction to the Moon. The pencil shape of this covariance matrix highlights the difficulty that optical sensors have in determining range.

4.9.2 Angle Between Horizon and Star

The variance of the scalar measurement of the angle between the horizon and a star is defined by

$$\mathbf{R}_\psi = E \left[\left(\tilde{\psi} - \psi \right)^2 \right] = \sigma_\psi^2 \quad (4.110)$$

where the angle ψ is defined as,

$$\cos \psi = \mathbf{e}_s^T \mathbf{e}_h \quad (4.111)$$

$$\cos \tilde{\psi} = \tilde{\mathbf{e}}_s^T \tilde{\mathbf{e}}_h \quad (4.112)$$

and \mathbf{e}_s is the line-of-sight unit vector to the star and \mathbf{e}_h is the line-of-sight unit vector to the closest point on the horizon. This notation is consistent with the definition from Eq. 4.94. Again, note that the frame in which \mathbf{e}_s and \mathbf{e}_h are

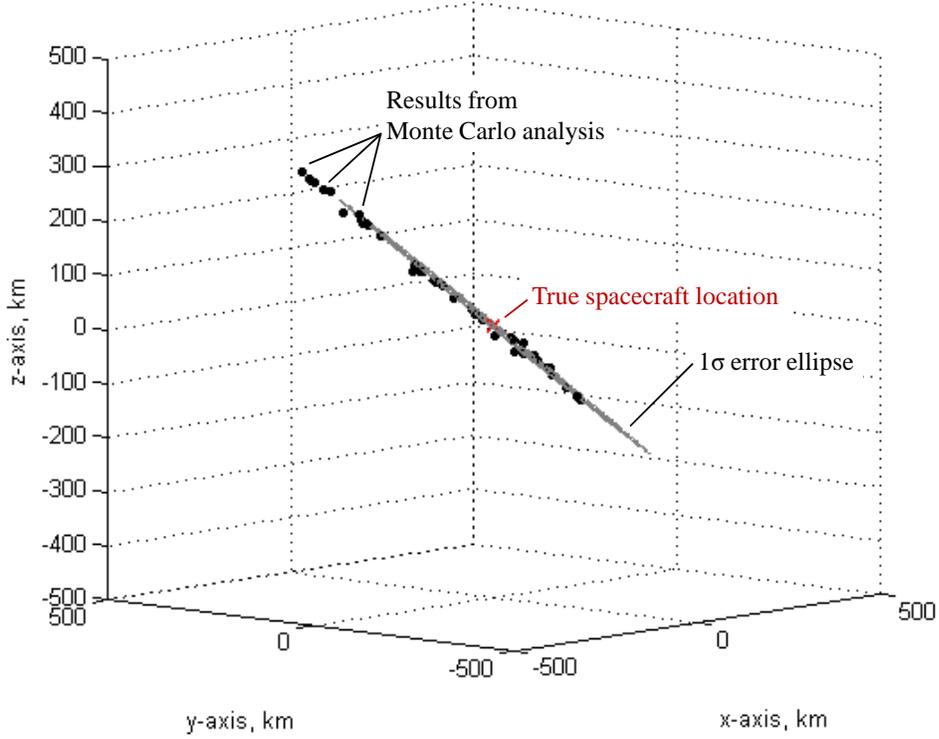


Figure 4.27: Error ellipse for instantaneous position fix from centroid and apparent diameter. Results for simulated image of Moon at a range of 200,636 km with a $1.5^\circ \times 1.5^\circ$ FOV camera.

expressed does not matter - here it is assumed that all vectors are expressed in the camera frame. Proceed by defining the the errors in the line-of-sight direction as

$$\boldsymbol{\epsilon}_s = \tilde{\mathbf{e}}_s - \mathbf{e}_s \quad \boldsymbol{\epsilon}_h = \tilde{\mathbf{e}}_h - \mathbf{e}_h \quad (4.113)$$

and the error in the measured angle between these vectors as

$$\delta\psi = \tilde{\psi} - \psi \quad (4.114)$$

Therefore, rewriting Eq. 4.112,

$$\cos (\psi + \delta\psi) = (\mathbf{e}_s + \boldsymbol{\epsilon}_s)^T (\mathbf{e}_h + \boldsymbol{\epsilon}_h) \quad (4.115)$$

Expanding both sides of this equation yields

$$\cos \psi \cos \delta\psi - \sin \psi \sin \delta\psi = \mathbf{e}_s^T \mathbf{e}_h + \mathbf{e}_s^T \boldsymbol{\epsilon}_h + \boldsymbol{\epsilon}_s^T \mathbf{e}_h + \boldsymbol{\epsilon}_s^T \boldsymbol{\epsilon}_h \quad (4.116)$$

If the error angle $\delta\psi$ is assumed to be small, then to first order

$$\mathbf{e}_s^T \mathbf{e}_h - \delta\psi \sin \psi = \mathbf{e}_s^T \mathbf{e}_h + \mathbf{e}_s^T \boldsymbol{\epsilon}_h + \boldsymbol{\epsilon}_s^T \mathbf{e}_h + \boldsymbol{\epsilon}_s^T \boldsymbol{\epsilon}_h \quad (4.117)$$

Therefore, solving for $\delta\psi$,

$$\delta\psi = -\frac{1}{\sin \psi} [\mathbf{e}_s^T \boldsymbol{\epsilon}_h + \boldsymbol{\epsilon}_s^T \mathbf{e}_h + \boldsymbol{\epsilon}_s^T \boldsymbol{\epsilon}_h] \quad (4.118)$$

Now, substituting this expression for $\delta\psi$ into Eq. 4.110 and assuming that $\boldsymbol{\epsilon}_s$ and $\boldsymbol{\epsilon}_h$ are uncorrelated,

$$\mathbf{R}_\psi = \frac{1}{\sin^2 \psi} \{ \mathbf{e}_s^T E [\boldsymbol{\epsilon}_h \boldsymbol{\epsilon}_h^T] \mathbf{e}_s + \mathbf{e}_h^T E [\boldsymbol{\epsilon}_s \boldsymbol{\epsilon}_s^T] \mathbf{e}_h \} \quad (4.119)$$

Both $\mathbf{R}_{C,h} = E [\boldsymbol{\epsilon}_h \boldsymbol{\epsilon}_h^T]$ and $\mathbf{R}_{C,s} = E [\boldsymbol{\epsilon}_s \boldsymbol{\epsilon}_s^T]$ are covariance matrices for a line-of-sight unit vector in the camera frame. The covariance for such measurements is given by the relation from Eq. 3.158. Therefore,

$$\mathbf{R}_\psi = \frac{1}{1 - (\mathbf{e}_s^T \mathbf{e}_h)^2} \{ \mathbf{e}_s^T \mathbf{R}_{C,h} \mathbf{e}_s + \mathbf{e}_h^T \mathbf{R}_{C,s} \mathbf{e}_h \} \quad (4.120)$$

Additionally, note that when using this measurement, errors associated with the current attitude estimate or camera misalignment do not enter into the measurement covariance (in contrast to the position fix from centroid and apparent diameter measurement).

4.10 Optical Navigation Measurement Sensitivity Matrices

Both of the measurement models being discussed may be written in the form

$$\tilde{\mathbf{y}} = h(\mathbf{x}) + \mathbf{v} \quad (4.121)$$

where $\tilde{\mathbf{y}}$ is the $m \times 1$ measurement, \mathbf{x} is the $n \times 1$ state vector, and \mathbf{v} is zero mean white noise. The measurement is assumed to be some nonlinear function of the state.

It is common practice to use a Taylor series expansion to linearize the measurement model about a reference solution. Taking the partial derivative of the measurement with respect to the state vector, yields the $m \times n$ measurement sensitivity matrix, \mathbf{H} . Errors are frequently assumed to behave in a linear fashion around the reference solution such that the measurement sensitivity matrix may be used to map errors in the state estimate to errors in the measurement. This approach is common in many types of filters (e.g. Extended Kalman Filter) and is discussed extensively in the literature.[150–152] Therefore,

$$\delta\mathbf{y} \approx \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \delta\mathbf{x} = \mathbf{H}\delta\mathbf{x} \quad (4.122)$$

where $\delta\mathbf{x}$ is the deviation of the state from the reference and $\delta\mathbf{y}$ is the deviation of the measurement from the reference.

4.10.1 Spacecraft Position from Centroid and Apparent Diameter

Construction of the measurement sensitivity matrix for the spacecraft position obtained from the centroid and apparent diameter measurement is straightforward. This particular measurement provides a direct measurement of the state corrupted by some measurement noise, $\mathbf{v}_{\mathbf{r}_c}$,

$$\tilde{\mathbf{y}} = h(\mathbf{x}) + \mathbf{v} = \mathbf{r}_c + \mathbf{v}_{\mathbf{r}_c} \quad (4.123)$$

Therefore, the measurement sensitivity matrix is just the identity matrix:

$$\mathbf{H} = \frac{\partial \mathbf{y}}{\partial \mathbf{r}_c} = \frac{\partial \mathbf{r}_c}{\partial \mathbf{r}_c} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.124)$$

4.10.2 Angle Between Horizon and Star

Finding the measurement sensitivity matrix for the angle between the horizon and a star is significantly more complex. In fact, the analytic computation of this matrix was deemed to be prohibitively complicated for the general case (although an approximation could be computed numerically). Therefore, the measurement sensitivity matrix derived here will be restricted to spherical (or nearly spherical) planets and moons. This, of course, does not mean that the projection of planet horizon is assumed to be a circle - the projection of a sphere is still an ellipse - and the navigation algorithms must still deal with ellipses on the image plane. This is a reasonable approximation for all of the planets, and most of the major moons, in our solar system. This derivation approach is based on some observations made by Battin in [24] and [153].

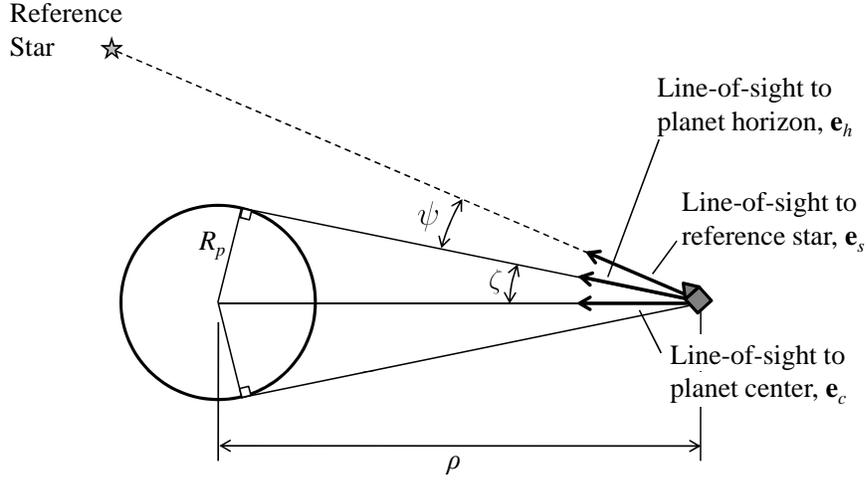


Figure 4.28: Geometry of star-horizon measurement for a spherical planet or moon.

Using the same nomenclature introduced in previous sections, begin by defining the measurement as

$$\tilde{\mathbf{y}} = h(\mathbf{x}) + \mathbf{v} = \psi + \delta\psi \quad (4.125)$$

so the measurement sensitivity matrix that must be derived here is given by

$$\mathbf{H} = \frac{\partial \mathbf{y}}{\partial \mathbf{r}_c} = \frac{\partial \psi}{\partial \mathbf{r}_c} \quad (4.126)$$

To compute $\partial\psi/\partial\mathbf{r}_c$, consider a spacecraft viewing a planet with the geometry shown in Fig. 4.28, where ζ is the planet apparent diameter and ψ is the angle between the horizon and the reference star. Therefore,

$$\cos(\psi + \zeta) = \mathbf{e}_s^T \mathbf{e}_c \quad (4.127)$$

where \mathbf{e}_s is the line-of-sight unit vector from the camera to the star and \mathbf{e}_c is the line-of-sight unit vector from the camera to the center of the planet.

As before, note that this expression is the same regardless of the frame being used. For this problem it is easiest to express these quantities in the inertial frame (or the same frame that the spacecraft position vector is expressed in). Therefore, recalling the relation from Eq. 4.78,

$$\cos(\psi + \zeta) = \mathbf{e}_s^T \left(\frac{-\mathbf{r}_c}{\rho} \right) = \mathbf{e}_s^T \frac{-\mathbf{r}_c}{\sqrt{\mathbf{r}_c^T \mathbf{r}_c}} \quad (4.128)$$

If the planet is assumed to be a sphere, then the apparent diameter is simply

$$\sin \zeta = \frac{R_p}{\rho} = \frac{R_p}{\sqrt{\mathbf{r}_c^T \mathbf{r}_c}} \quad (4.129)$$

where R_p is the radius of the planet. Taking the derivative of Eq. 4.128 and Eq. 4.129 with respect to the camera position vector, \mathbf{r}_c , will yield the following two expressions

$$-\sin(\psi + \zeta) \left[\frac{\partial \psi}{\partial \mathbf{r}_c} + \frac{\partial \zeta}{\partial \mathbf{r}_c} \right] = \mathbf{e}_s^T \left[-\frac{1}{\rho} \mathbf{I}_{3 \times 3} + \frac{1}{\rho^3} \mathbf{r}_c \mathbf{r}_c^T \right] \quad (4.130)$$

$$\cos \zeta \frac{\partial \zeta}{\partial \mathbf{r}_c} = -\frac{R_p}{\rho^3} \mathbf{r}_c^T \quad (4.131)$$

Now, substituting $\partial \zeta / \partial \mathbf{r}_c$ from Eq. 4.131 into Eq. 4.130, and solving for $\partial \psi / \partial \mathbf{r}_c$

$$\frac{\partial \psi}{\partial \mathbf{r}_c} = \frac{1}{\sin(\psi + \zeta)} \mathbf{e}_s^T \left[\frac{1}{\rho} \mathbf{I}_{3 \times 3} - \frac{1}{\rho^3} \mathbf{r}_c \mathbf{r}_c^T \right] + \frac{R_p}{\rho^3 \cos \zeta} \mathbf{r}_c^T \quad (4.132)$$

Therefore, the measurement sensitivity matrix for the star-horizon measurement is given by

$$\mathbf{H} = \frac{1}{\sin(\psi + \zeta)} \mathbf{e}_s^T \left[\frac{1}{\rho} \mathbf{I}_{3 \times 3} - \frac{1}{\rho^3} \mathbf{r}_c \mathbf{r}_c^T \right] + \frac{R_p}{\rho^3 \cos \zeta} \mathbf{r}_c^T \quad (4.133)$$

Chapter 5

Optical Attitude Estimation

Attitude estimation is a critical first step for an autonomous optical navigation system. Recall that the image processing and measurement algorithms from Chapter 4 require an estimate of the spacecraft attitude. Fortunately, the fundamental issues regarding optical attitude estimate are well known. This chapter begins with a brief summary of existing attitude estimation and filtering techniques. Next, because attitude filtering requires knowledge of the dynamics, key results from attitude dynamics are presented - primarily from the standpoint of introducing the notation to be used in subsequent sections.

With the preliminaries established, the focus of the chapter then turns to the development of a new nonlinear attitude filter: the Sequential Optimal Attitude Recursion (SOAR) Filter. This discussion begins by reviewing the classic solution to the Wahba problem and representations of attitude covariance, which forms the starting point for the development of the SOAR Filter. The remaining sections discuss the complete derivation of the SOAR Filter and provide a comparison between this new filter and existing attitude filters.

5.1 Existing Attitude Filtering Algorithms

The problem of estimating spacecraft attitude from a set of unit vector observations has received much attention since the 1960s. The most well known formulation of this problem was given by Grace Wahba in 1965[154] and is commonly referred to as the “Wahba Problem.” If the Wahba Problem is written in terms of the attitude quaternion, an analytic solution exists in the form of the solution to an eigenvalue-eigenvector problem. The optimal attitude is simply given by the eigenvector associated with the largest eigenvalue.[155] Numerous algorithms have been developed over the years to find fast and robust solutions to this problem.[156] Some of the most well known algorithms include the QUaternion ESTimator (QUEST)[157], the ESTimator of the Optimal Quaternion (ESOQ),[158, 159] and Singular Value Decomposition (SVD).[160] Because the models created in this work use the QUEST algorithm to generate the quaternion measurements produced by star trackers, the complete derivation of the QUEST algorithm is reproduced in a nomenclature consistent with this work in Appendix D.

Traditional solutions to the Wahba Problem are batch estimators that assume all the unit vector observations occur at the same time. The Wahba Problem is a weighted least squares problem. If the weights in this problem are chosen to be the inverse of the measurement noise variance, the result is a maximum likelihood estimate of the attitude. The objective of the present research is to extend the traditional Wahba Problem into a framework that allows for the creation of an optimal attitude filter.

One of the most common methods of attitude filtering in modern systems is the Multiplicative Extended Kalman Filter (MEKF). This method uses the four-component attitude quaternion to represent the attitude, but a three-component representation of the attitude in the filter. The MEKF is structured this way because while the attitude quaternion is globally nonsingular, making it a good choice for representing spacecraft attitude, it must obey a unity norm constraint, making direct implementation of the quaternion in a normal Kalman filter difficult. Additionally, a small angle assumption allows the three-component attitude representation to be cast in a form consistent with the additive nature of the Kalman filter (i.e. $\mathbf{x} = \hat{\mathbf{x}} + \delta\mathbf{x}$).

Unlike the MEKF, sequential Wahba Problem filters estimate the full quaternion without requiring solutions to be computed as small angle deviations from a reference attitude. The batch solutions to the Wahba Problem have also been shown to be extremely robust. These are some of the advantages that have led researchers to investigate sequential solutions to the Wahba Problem for attitude filtering. Shuster provided the earliest known work in this topic with the development of “Filter QUEST” in a number of articles.[161, 162] Subsequently, Bar-Itzhack introduced an alternate sequentialization approach known as the REQUEST algorithm.[163] Several years after the introduction of the REQUEST algorithm, Shuster[164] demonstrated that although the Filter QUEST and REQUEST algorithms approach the problem from different perspectives, they are mathematically equivalent. Both Filter QUEST and REQUEST algorithms are examples of suboptimal fading memory

filters. Shuster demonstrates[161, 162] that under a specific set of conditions, the optimal fading memory factor, α_k , may be analytically computed. The equation for α_k is simple, fast to compute, and provides an excellent approximation of the more general optimal fading memory factor for many practical cases. The assumptions required to arrive at this result are rarely met, however, and a better value of α_k may be computed at the expense of a little more computation time.

An approach for the optimal blending of the new attitude measurements with old attitude estimates directly in the Wahba Problem solution framework was proposed in 2004 by Choukroun with the Optimal-REQUEST algorithm.[165] This algorithm, however, requires access to individual unit vector measurements. This may cause difficulties if an instrument provides a measurement in the form of an attitude quaternion, as is the case with most commercially available star trackers. Additionally, like Filter QUEST and REQUEST, Optimal-REQUEST is only capable of estimating attitude. None of the sequential Wahba Problem filters discussed above are capable of incorporating estimates of other parameters, such as sensor biases.

Taking a different approach, Psiaki developed the Extended-QUEST algorithm.[166] The Extended-QUEST algorithm offers an optimal estimate of the attitude and is capable of also estimating non-attitude states. This algorithm is formulated as a square-root information filter that contains a propagation phase and a state update phase. The formulation of Extended-QUEST requires the QR factorization of the propagated information matrix for use

in the filter. It also requires the solution to a more general objective function than required for Filter QUEST or REQUEST. Despite these increases in complexity and computations, Extended-QUEST provides an optimal estimate of both attitude and non-attitude states. It has been shown to be robust with convergence properties that are substantially better than seen in the MEKF.[166]

Other algorithms have also been developed that are capable of estimating both attitude states and non-attitude states. Most notable among these is an algorithm introduced by Markley.[167, 168] While many other nonlinear attitude filtering approaches exist, they are not discussed here. Thorough discussions of these other methods may be found in the literature.[169]

It is within this context that a new nonlinear attitude filtering algorithm is proposed: the Sequential Optimal Attitude Recursion (SOAR) Filter. The derivation of the SOAR Filter begins by creating a maximum likelihood estimate (MLE) of the state vector through Bayesian estimation. A fresh look at the sequentialization of the Wahba Problem allows the classic Wahba Problem to be recast in an MLE framework. Additionally, the Bayesian estimation approach allows for the seamless inclusion of non-attitude states in the SOAR Filter. Using ideas from the Extended Kalman Filter (EKF), the covariance inflation associated with the propagation of the state is examined. Treating the covariance in this manner allows for the straightforward inclusion of process noise and addresses the problem of the suboptimal fading memory factor seen in Filter QUEST and REQUEST. If the attitude covariance and measure-

ment covariance are known at the time of the update, the optimal estimate of the attitude may be determined.

The SOAR Filter shares many similarities with some of the previous attitude filtering methods. As subsequent discussions will show, the SOAR Filter is most similar to Extended-QUEST.

5.2 Attitude Dynamics

This section provides a brief review of some key kinematic relations from attitude dynamics. More detailed information on this topic may be found in [78] and [170]. Define the time derivative of the DCM, $\dot{\mathbf{T}}$, as

$$\dot{\mathbf{T}} \equiv \lim_{\Delta t \rightarrow 0} \frac{\mathbf{T}(t + \Delta t) - \mathbf{T}(t)}{\Delta t} \quad (5.1)$$

Assuming that a rotation of $\Delta\boldsymbol{\theta}$ occurs during the time Δt and applying the small angle approximation of Eq. 3.5

$$\dot{\mathbf{T}} = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} (\mathbf{I}_{3 \times 3} - [\Delta\boldsymbol{\theta} \times]) \mathbf{T}(t) - \mathbf{T}(t) = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} (-[\Delta\boldsymbol{\theta} \times]) \mathbf{T}(t) \quad (5.2)$$

Recalling that $\boldsymbol{\omega}$ is defined as $\boldsymbol{\omega} \equiv \lim_{\Delta t \rightarrow 0} \Delta\boldsymbol{\theta}/\Delta t$,

$$\dot{\mathbf{T}} = -[\boldsymbol{\omega} \times] \mathbf{T} \quad (5.3)$$

Similarly, if the time derivative of the attitude quaternion, $\dot{\bar{\mathbf{q}}}$, is defined as

$$\dot{\bar{\mathbf{q}}} \equiv \lim_{\Delta t \rightarrow 0} \frac{\bar{\mathbf{q}}(t + \Delta t) - \bar{\mathbf{q}}(t)}{\Delta t} \quad (5.4)$$

then the following kinematic relations are known to be true:[81]

$$\dot{\bar{\mathbf{q}}} = \frac{1}{2} \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \bar{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Omega} \bar{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Xi} \boldsymbol{\omega} \quad (5.5)$$

where the matrices $\mathbf{\Omega}$ and \mathbf{Q} are given by:

$$\mathbf{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -[\boldsymbol{\omega} \times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \quad (5.6)$$

$$\mathbf{\Xi}(\bar{\mathbf{q}}) = \begin{bmatrix} q_4 \mathbf{I}_{3 \times 3} + [\mathbf{q} \times] \\ -\mathbf{q}^T \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (5.7)$$

Additionally, if rigid body attitude dynamics are assumed, the angular acceleration is given by Euler's equations:

$$\dot{\boldsymbol{\omega}} = \mathbf{J}_{cm}^{-1} (\mathbf{m}_{ext} - [\boldsymbol{\omega} \times] \mathbf{J}_{cm} \boldsymbol{\omega}) \quad (5.8)$$

where \mathbf{J}_{cm} is the inertia matrix for the spacecraft expressed in the body frame and \mathbf{m}_{ext} is the sum of the external moments applied to the spacecraft.

5.3 A Review of the Classical Wahba Problem

The problem of estimating vehicle attitude from a set of unit vector observations has been studied in great detail. The objective function commonly used to describe this problem was posed by Grace Wahba in 1965 and is frequently referred to as the ‘‘Wahba Problem.’’ [154] The Wahba Problem statement, using traditional nomenclature, is given by:

$$\text{Min } J(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^m w_i \|\mathbf{w}_i - \mathbf{A} \mathbf{v}_i\|^2 \quad (5.9)$$

where m is the number of observed unit vectors, w_i is a positive weighting on the i -th observation, \mathbf{w}_i is the measured unit vector for the i -th observation

as expressed in the body frame, \mathbf{v}_i is the known reference unit vector for the i -th observation in a known inertial frame, and \mathbf{A} is the rotation matrix that transforms a vector from the inertial frame to the body frame.

For the purposes of this discussion, however, a slightly different notation will be used. First, the rotation matrix that transforms a vector from the inertial frame to the body frame will be denoted by \mathbf{T}_B^I . Therefore, the matrix \mathbf{T}_B^I describes the attitude of the vehicle body frame, B , with respect to the inertial frame, I . Second, note that the unit vectors \mathbf{w}_i and \mathbf{v}_i in Eq. 5.9 are actually the same unit vector expressed in two different frames. Therefore, denote this unit vector as \mathbf{e}_i , such that $(\mathbf{e}_i)_I$ is the reference unit vector expressed in the inertial frame, i.e. $(\mathbf{e}_i)_I = \mathbf{v}_i$. Unfortunately, measurement errors prevent \mathbf{w}_i from simply being \mathbf{e}_i expressed in the actual body frame. Therefore, let $(\tilde{\mathbf{e}}_i)_B$ be the perturbed (noisy) i -th unit vector expressed in the body frame, \tilde{B}_i , that yields the measured direction of an observed object, i.e. $(\tilde{\mathbf{e}}_i)_B = \mathbf{w}_i$. Although this notation is more cumbersome, it helps emphasize the frames and vectors in a more explicit fashion. Therefore the Wahba Problem may be rewritten in this new notation as

$$\text{Min } J(\mathbf{T}_B^I) = \frac{1}{2} \sum_{i=1}^m w_i \left\| (\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I (\mathbf{e}_i)_I \right\|^2 \quad (5.10)$$

In order to proceed, Wahba's problem must be manipulated for implementation in an attitude determination algorithm. Returning to Eq. 5.10, the

norm in the objective function may be expanded as follows:

$$\begin{aligned} \text{Min } J(\mathbf{T}_B^I) = \frac{1}{2} \sum_{i=1}^m w_i \left[(\tilde{\mathbf{e}}_i)_B^T (\tilde{\mathbf{e}}_i)_B - 2 (\tilde{\mathbf{e}}_i)_B^T \mathbf{T}_B^I (\mathbf{e}_i)_I \right. \\ \left. + (\mathbf{T}_B^I (\mathbf{e}_i)_I)^T (\mathbf{T}_B^I (\mathbf{e}_i)_I) \right] \end{aligned} \quad (5.11)$$

recognizing that $(\tilde{\mathbf{e}}_i)_B$ and $\mathbf{T}_B^I (\mathbf{e}_i)_I$ are unit vectors, Eq. 5.11 may be simplified to

$$\text{Min } J(\mathbf{T}_B^I) = \sum_{i=1}^m w_i - \sum_{i=1}^m w_i (\tilde{\mathbf{e}}_i)_B^T \mathbf{T}_B^I (\mathbf{e}_i)_I \quad (5.12)$$

Let the arbitrary constant $\sum_{i=1}^m w_i = \lambda_0$. Shuster was the first to demonstrate that a selection of $w_i = 1/\sigma_{\phi,i}^2$ yields the maximum likelihood estimate.[120]

This leads to the following modification of the objective function

$$\text{Min } J(\mathbf{T}_B^I) = \lambda_0 - \sum_{i=1}^m w_i \text{tr} \left[(\tilde{\mathbf{e}}_i)_B^T \mathbf{T}_B^I (\mathbf{e}_i)_I \right] \quad (5.13)$$

Now, recalling that one may cyclically permute multiplication within the trace operator, the above may be further rewritten as is commonly seen in the literature:[156]

$$\text{Min } J(\mathbf{T}_B^I) = \lambda_0 - \sum_{i=1}^m w_i \text{tr} \left[\mathbf{T}_B^I (\mathbf{e}_i)_I (\tilde{\mathbf{e}}_i)_B^T \right] = \lambda_0 - \text{tr} \left[\mathbf{T}_B^I \mathbf{B}^T \right] \quad (5.14)$$

where \mathbf{B} is called the attitude profile matrix and is given by

$$\mathbf{B} = \sum_{i=1}^m w_i (\tilde{\mathbf{e}}_i)_B (\mathbf{e}_i)_I^T \quad (5.15)$$

Continue the rewriting of the objective function by letting $g(\mathbf{T}_B^I) = \text{tr} \left[\mathbf{T}_B^I \mathbf{B}^T \right]$, yielding:

$$\text{Min } J(\mathbf{T}_B^I) = \lambda_0 - g(\mathbf{T}_B^I) \quad (5.16)$$

Recalling the relationship between the rotation matrix and the corresponding attitude quaternion from Eq. 3.11, it can be shown that

$$g(\mathbf{T}_B^I) = g(\mathbf{T}_B^I(\bar{\mathbf{q}})) = h(\bar{\mathbf{q}}) \quad (5.17)$$

This allows for the rewriting of Eq. 5.16 in terms of the quaternion instead of the attitude matrix

$$\text{Min } J(\bar{\mathbf{q}}) = \lambda_0 - h(\bar{\mathbf{q}}) \quad (5.18)$$

Equation 3.11 also suggests that $h(\bar{\mathbf{q}})$ is quadratic in $\bar{\mathbf{q}}$. To verify this observation, and to obtain an expression for the objective function in terms of the quaternion, use the identity from Eq. 3.11 to rewrite Eq. 5.17 as

$$\begin{aligned} g(\mathbf{T}_B^I) &= \text{tr}[\mathbf{T}_B^I \mathbf{B}^T] = h(\bar{\mathbf{q}}) \\ &= \text{tr}[[(q_4^2 - \mathbf{q}^T \mathbf{q}) \mathbf{I}_{3 \times 3} + 2\mathbf{q}\mathbf{q}^T - 2q_4[\mathbf{q} \times]] \mathbf{B}^T] \end{aligned} \quad (5.19)$$

Distributing \mathbf{B}^T and the trace operator,

$$h(\bar{\mathbf{q}}) = (q_4^2 - \mathbf{q}^T \mathbf{q}) \text{tr}[\mathbf{B}^T] + 2\text{tr}[\mathbf{q}\mathbf{q}^T \mathbf{B}^T] - 2q_4 \text{tr}[[\mathbf{q} \times] \mathbf{B}^T] \quad (5.20)$$

If the new variable μ is defined as the trace of the attitude profile matrix \mathbf{B} ,

$$\mu = \text{tr}[\mathbf{B}] \quad (5.21)$$

then $h(\bar{\mathbf{q}})$ may be rewritten as

$$h(\bar{\mathbf{q}}) = (q_4^2 - \mathbf{q}^T \mathbf{q}) \mu + 2\text{tr}[\mathbf{q}\mathbf{q}^T \mathbf{B}^T] - 2q_4 \text{tr}[[\mathbf{q} \times] \mathbf{B}^T] \quad (5.22)$$

Further, cyclically permute the values within the trace of the second term,

$$h(\bar{\mathbf{q}}) = (q_4^2 - \mathbf{q}^T \mathbf{q}) \mu + 2\text{tr}[\mathbf{q}^T \mathbf{B}^T \mathbf{q}] - 2q_4 \text{tr}[[\mathbf{q} \times] \mathbf{B}^T] \quad (5.23)$$

the second term is now a scalar and the trace operator may be dropped.

Further recognize that $2\mathbf{q}^T\mathbf{B}^T\mathbf{q} = \mathbf{q}^T\mathbf{S}\mathbf{q}$ where

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^T \quad (5.24)$$

Therefore, again rewrite $h(\bar{\mathbf{q}})$ as

$$h(\bar{\mathbf{q}}) = (q_4^2 - \mathbf{q}^T\mathbf{q})\mu + \mathbf{q}^T\mathbf{S}\mathbf{q} - 2q_4\text{tr}[[\mathbf{q}\times]\mathbf{B}^T] \quad (5.25)$$

Now turn attention to the final term. Expand the trace and collect in terms of elements in \mathbf{q} :

$$\text{tr}[[\mathbf{q}\times]\mathbf{B}^T] = (b_{32} - b_{23})q_1 + (b_{13} - b_{31})q_2 + (b_{21} - b_{12})q_3 = -\mathbf{z}^T\mathbf{q} \quad (5.26)$$

where

$$[\mathbf{z}\times] = \mathbf{B}^T - \mathbf{B} \quad (5.27)$$

Recognizing that $\mathbf{z}^T\mathbf{q}$ is a scalar and $\mathbf{z}^T\mathbf{q} = \mathbf{q}^T\mathbf{z}$,

$$h(\bar{\mathbf{q}}) = (q_4^2 - \mathbf{q}^T\mathbf{q})\mu + \mathbf{q}^T\mathbf{S}\mathbf{q} + q_4\mathbf{z}^T\mathbf{q} + q_4\mathbf{q}^T\mathbf{z} \quad (5.28)$$

Breaking out a $\bar{\mathbf{q}}^T = [\mathbf{q}^T q_4]$ to the left,

$$h(\bar{\mathbf{q}}) = \begin{bmatrix} \mathbf{q}^T & q_4 \end{bmatrix} \begin{bmatrix} -\mu\mathbf{q} + \mathbf{S}\mathbf{q} + q_4\mathbf{z} \\ \mu q_4 + \mathbf{z}^T\mathbf{q} \end{bmatrix} \quad (5.29)$$

Now factor a $\bar{\mathbf{q}}$ to the right,

$$h(\bar{\mathbf{q}}) = \begin{bmatrix} \mathbf{q}^T & q_4 \end{bmatrix} \begin{bmatrix} \mathbf{S} - \mu\mathbf{I}_{3\times 3} & \mathbf{z} \\ \mathbf{z}^T & \mu \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \quad (5.30)$$

Substituting this result for $h(\bar{\mathbf{q}})$ back into Eq. 5.18 leads to the well known result[78, 155, 157]

$$\text{Min } J(\bar{\mathbf{q}}) = \lambda_0 - h(\bar{\mathbf{q}}) = \lambda_0 - \bar{\mathbf{q}}^T\mathbf{K}\bar{\mathbf{q}} \quad (5.31)$$

where \mathbf{K} is called the Davenport matrix and is defined as

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \mu \mathbf{I}_{3 \times 3} & \mathbf{z} \\ \mathbf{z}^T & \mu \end{bmatrix} \quad (5.32)$$

where

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^T \quad (5.33)$$

$$\mu = \text{tr} [\mathbf{B}] \quad (5.34)$$

$$[\mathbf{z} \times] = \mathbf{B}^T - \mathbf{B} \quad \text{or} \quad \mathbf{z} = \sum_{i=1}^m w_i (\tilde{\mathbf{e}}_i)_B \times (\mathbf{e}_i)_I \quad (5.35)$$

The attitude that minimizes the objective function in Eq. 5.31 may be found by the straightforward application of concepts from optimal control theory.[171] Begin by recalling the quaternion unity norm constraint $\|\bar{\mathbf{q}}\| = 1$ and adjoin this constraint to the objective function through a Lagrange multiplier, λ . Also, write the objective function in terms of the quaternion, $\bar{\mathbf{q}}$, instead of the DCM, \mathbf{T}_B^I , as follows from Eq. 5.17 and Eq. 5.31:

$$\text{Min} \quad \mathcal{J}(\bar{\mathbf{q}}, \lambda) = \lambda_0 - \bar{\mathbf{q}}^T \mathbf{K} \bar{\mathbf{q}} + \lambda (\bar{\mathbf{q}}^T \bar{\mathbf{q}} - 1) \quad (5.36)$$

The optimal attitude is found by setting the first differential of $\mathcal{J}(\bar{\mathbf{q}}, \lambda)$ to zero. The optimal attitude is given by the solution to[78, 155, 157]

$$\mathbf{K} \bar{\mathbf{q}} = \lambda \bar{\mathbf{q}} \quad (5.37)$$

Note that Eq. 5.37 is an eigenvalue-eigenvector problem. It is from this point that various solutions have emerged. Despite all the discussion circulating about the various attitude determination algorithms, most of these are simply different methods for solving the eigenvalue-eigenvector problem in Eq.

5.37. Davenport's q-Method, for example, directly solves Eq. 5.37 as if it were an ordinary eigenvalue-eigenvector problem,[156] thus making it one of the more robust algorithms because it relies on relatively robust but slow numerical methods for solving the eigenvalue-eigenvector problem. Further, by substituting Eq. 5.37 into Eq. 5.31, the original objective function becomes:

$$J(\bar{\mathbf{q}}) = \lambda_0 - \bar{\mathbf{q}}^T \lambda \bar{\mathbf{q}} = \lambda_0 - \lambda \bar{\mathbf{q}}^T \bar{\mathbf{q}} = \lambda_0 - \lambda \quad (5.38)$$

Therefore, it may be noted that the optimal attitude is achieved with the largest eigenvalue.

The QUEST algorithm,[157] developed by Shuster, solves this problem by Newton-Raphson iteration on the characteristic equation to find only the largest eigenvalue and associated eigenvector. A full derivation of the QUEST algorithm using notation consistent with this work is provided in Appendix D. While faster than Davenport's q-Method, the solution approach introduces a singularity for a 180 degree rotation that must be addressed in the algorithm implementation (typically through logic checks and sequential rotations).

ESTimators of the Optimal Quaternion (ESOQ and ESOQ-2),[158, 159] developed by Mortari, are also fundamentally based on the eigenvalue problem described in Eq. 5.37. Recognizing that

$$(\mathbf{K} - \lambda \mathbf{I}_{4 \times 4}) \bar{\mathbf{q}} = \mathbf{H} \bar{\mathbf{q}} = \mathbf{0} \quad (5.39)$$

it is clear that $\bar{\mathbf{q}}$ must lie in the null space of \mathbf{H} . Further if $\mathbf{H}^T = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3 \ \mathbf{h}_4]$, then $\bar{\mathbf{q}}$ must be perpendicular to all \mathbf{h}_i . From here, the ESOQ algorithm

computes the optimal quaternion through a 4-dimensional cross-product operation (a special case of the n -dimensional cross-product)[172] between two vectors in \mathbf{H} : \mathbf{h}_i and \mathbf{h}_j , $i \neq j$. In the same year that ESOQ was introduced, Mortari also proposed a follow-on algorithm called ESOQ-2. This algorithm starts with Eq. 5.37 and computes a symmetric 3×3 matrix of rank 2, \mathbf{M} , such that $\mathbf{M}\mathbf{e}_\theta = \mathbf{0}$. Using a similar approach to that seen in the original ESOQ, the attitude may now be determined by taking the regular 3-dimensional cross product of two rows of \mathbf{M} .

5.4 Attitude Error and Covariance Relations

5.4.1 The Attitude Error Quaternion

Let the error in the attitude be described by the small error quaternion, $\delta\bar{\mathbf{q}}$:

$$\delta\bar{\mathbf{q}} = \bar{\mathbf{q}} \otimes \hat{\bar{\mathbf{q}}}^{-1} \quad (5.40)$$

where \otimes denotes the quaternion product operator. Physically, $\delta\bar{\mathbf{q}}$ may be interpreted as the quaternion that rotates the best estimated attitude to the true attitude. If one assumes small angles,

$$\delta\bar{\mathbf{q}} = \begin{bmatrix} \delta\mathbf{q} \\ \delta q_4 \end{bmatrix} \approx \begin{bmatrix} \delta\boldsymbol{\theta}/2 \\ 1 \end{bmatrix} \quad (5.41)$$

where $\delta\boldsymbol{\theta}$ is a three-dimensional parameterization of the attitude given by $\mathbf{e}_{\delta\theta}\delta\theta$. Here, $\mathbf{e}_{\delta\theta}$ is a unit vector denoting the axis of rotation and $\delta\theta$ is the magnitude of the rotation about that axis in radians. Therefore, $2\delta\mathbf{q} \approx \delta\boldsymbol{\theta}$, which yields

$$\mathbf{P}_{\theta\theta} = E[\delta\boldsymbol{\theta}\delta\boldsymbol{\theta}^T] = 4E[\delta\mathbf{q}\delta\mathbf{q}^T] \quad (5.42)$$

This choice means that $\mathbf{P}_{\theta\theta}$ is the attitude covariance matrix as expressed in the body frame. Additionally, assume that the quaternion attitude estimate is unbiased:

$$E[\delta\bar{\mathbf{q}}] = E\left[\begin{bmatrix} \delta\mathbf{q} \\ \delta q_4 \end{bmatrix}\right] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (5.43)$$

The definition in Eq. 5.40 also means that the small angle error vector is defined such that

$$\mathbf{T}(\boldsymbol{\theta}) = \mathbf{T}(\delta\boldsymbol{\theta})\mathbf{T}(\hat{\boldsymbol{\theta}}) \quad (5.44)$$

If the three-dimensional parameterization of the attitude given by $\boldsymbol{\theta}$ is defined with respect to the attitude estimate given by $\hat{\mathbf{q}}$, then $\hat{\boldsymbol{\theta}} = \mathbf{0}$ and $\mathbf{T}(\hat{\boldsymbol{\theta}}) = \mathbf{I}_{3 \times 3}$ by definition. Therefore,

$$\delta\boldsymbol{\theta} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} = \boldsymbol{\theta} - \mathbf{0} = \boldsymbol{\theta} \quad (5.45)$$

5.4.2 Important Attitude Covariance Relations

Next, a few key relations that relate the attitude profile matrix to the covariance matrix are derived. Begin by recalling the Cramér-Rao inequality,[173] which states

$$\mathbf{P}_{\mathbf{xx}} = E\left[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\right] \geq (\mathcal{F}_{\mathbf{xx}})^{-1} \quad (5.46)$$

where $\mathcal{F}_{\mathbf{xx}}$ is the Fisher information matrix. The Fisher information matrix is further defined as

$$\mathcal{F}_{\mathbf{xx}} = E\left[\frac{\partial^2 J(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}}\right] \quad (5.47)$$

where the scalar objective function J is the negative log-likelihood function (i.e. $J(\mathbf{x}) = -\ln p(\mathbf{x}|\mathbf{y})$, where $p(\mathbf{x}|\mathbf{y})$ is the probability of observing the state \mathbf{x} given the observations \mathbf{y}). The equality in Eq. 5.46 is obtained if and only if $p(\mathbf{x}|\mathbf{y})$ is Gaussian and the following are both true[173]

$$\left[\frac{\partial}{\partial \mathbf{x}} J(\mathbf{x}) \right]^T = - \left[\frac{\partial}{\partial \mathbf{x}} \ln p(\mathbf{x}|\mathbf{y}) \right]^T = \mathbf{C} [\mathbf{x} - \hat{\mathbf{x}}] \quad (5.48)$$

where \mathbf{C} is a matrix that is independent of \mathbf{x} and \mathbf{y} . If the condition of Eq. 5.48 is applied to a Gaussian $p(\mathbf{x}|\mathbf{y})$, then it may be shown that the observations must be a linear function of \mathbf{x} . This observation was also made by Shuster in [120]. Therefore, if these conditions are both satisfied then $\mathcal{F}_{\mathbf{xx}} = \mathbf{P}_{\mathbf{xx}}^{-1}$. Unfortunately, this is not the case in the present problem. When the above conditions are not met, $\mathcal{F}_{\mathbf{xx}}$ approaches $\mathbf{P}_{\mathbf{xx}}^{-1}$ as the number of measurements becomes infinite:[120]

$$\mathbf{P}_{\mathbf{xx}}^{-1} = \lim_{n \rightarrow \infty} \mathcal{F}_{\mathbf{xx}} \quad (5.49)$$

where n is the number of observations.

To be consistent with earlier notation, let $\delta\boldsymbol{\theta}$ be the small angle rotation that rotates the estimate of the vehicle attitude into the true attitude. Therefore, assuming small angles,[82]

$$\mathbf{T}_B^I = \exp \{[-\delta\boldsymbol{\theta} \times]\} \mathbf{T}_B^I \quad (5.50)$$

Using this relation, the Wahba Problem objective function of Eq. 5.14 evaluated at the true attitude may be rewritten as a function of $\delta\boldsymbol{\theta}$

$$J(\delta\boldsymbol{\theta}) = \lambda_0 - \text{tr} [\exp \{[-\delta\boldsymbol{\theta} \times]\} \mathbf{T}_B^I \mathbf{B}^T] \quad (5.51)$$

Expanding the matrix exponential of $[-\delta\boldsymbol{\theta}\times]$ to second order (because the equation for $J(\delta\boldsymbol{\theta})$ must be differentiated twice),

$$J(\delta\boldsymbol{\theta}) = \lambda_0 - \text{tr} \left[\left(\mathbf{I}_{3\times 3} + [-\delta\boldsymbol{\theta}\times] + \frac{1}{2} [-\delta\boldsymbol{\theta}\times]^2 \right) \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \right] \quad (5.52)$$

Because the Wahba problem has been shown to be equivalent to the negative log-likelihood function (to first order) if the measurement covariance is chosen to be Eq. 3.141,[120] the Fisher information matrix may be computed by taking the partial derivative of $J(\delta\boldsymbol{\theta})$ twice with respect to $\delta\boldsymbol{\theta}$:

$$\mathcal{F}_{\theta\theta} = \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \mathbf{I}_{3\times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (5.53)$$

The details of this derivation are provided in Appendix E. Therefore,

$$\mathbf{P}_{\theta\theta}^{-1} \approx \mathcal{F}_{\theta\theta} = \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \mathbf{I}_{3\times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (5.54)$$

Rearranging the above equation (details also in Appendix E) for the attitude profile matrix \mathbf{B} ,

$$\mathbf{B} = \left[\frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3\times 3} - \mathcal{F}_{\theta\theta} \right] \mathbf{T}_{\hat{B}}^I \quad (5.55)$$

The relationships of Eq. 5.54 and Eq. 5.55 were first shown by Shuster in 1989.[120] From Eq. 5.54 and Eq. 5.55, one may freely move back and forth between \mathbf{B} and $\mathbf{P}_{\theta\theta}$.

5.4.3 The Wahba Problem Objective Function in Terms of Fisher Information

In preparation for the development of the SOAR algorithm, it is useful to find the value of the Wahba Problem objective function in terms of the

Fisher information matrix. Begin by substituting Eq. 5.55 into Eq. 5.52

$$J(\boldsymbol{\delta\theta}) = \lambda_0 - \text{tr} \left[\left(\mathbf{I}_{3 \times 3} + [-\boldsymbol{\delta\theta} \times] + \frac{1}{2} [-\boldsymbol{\delta\theta} \times]^2 \right) \mathbf{T}_{\hat{B}}^I \mathbf{T}_I^{\hat{B}} \left(\frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3 \times 3} - \mathcal{F}_{\theta\theta}^T \right) \right] \quad (5.56)$$

Recognizing that $\mathcal{F}_{\theta\theta}$ is symmetric, and expanding the terms in the objective function yields

$$J(\boldsymbol{\delta\theta}) = \lambda_0 - \frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] + \frac{1}{2} \boldsymbol{\delta\theta}^T \mathcal{F}_{\theta\theta} \boldsymbol{\delta\theta} \quad (5.57)$$

Taking closer look at the second term and substituting Eq. 5.54 for $\mathcal{F}_{\theta\theta}$,

$$\frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] = \frac{1}{2} \text{tr} [\text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \mathbf{I}_{3 \times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T] = \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] = \hat{\mathbf{q}}^T \mathbf{K} \hat{\mathbf{q}} \quad (5.58)$$

Therefore, substituting this into Eq. 5.57 means that the Wahba Problem objective function evaluated at the true attitude may be written in terms of the Fisher information matrix as

$$J(\boldsymbol{\delta\theta}) = \lambda_0 - \hat{\mathbf{q}}^T \mathbf{K} \hat{\mathbf{q}} + \frac{1}{2} \boldsymbol{\delta\theta}^T \mathcal{F}_{\theta\theta} \boldsymbol{\delta\theta} \quad (5.59)$$

Looking again at Eq. 5.38, it follows that

$$J(\boldsymbol{\delta\theta}) = J(\hat{\mathbf{q}}) + \frac{1}{2} \boldsymbol{\delta\theta}^T \mathcal{F}_{\theta\theta} \boldsymbol{\delta\theta} \quad (5.60)$$

The form of Eq. 5.60 is not surprising given that the solution to the Wahba problem is known to produce a maximum likelihood estimate of the attitude. It is also interesting to note that the difference in the value of the objective function at the true attitude and the value of the objective function at the best estimate of the attitude is equivalent to the square of the Mahalanobis distance[174, 175] between these two attitudes.

The relation in Eq. 5.60 allows for a number of interesting observations. First, this relation will be critical in the derivation of the SOAR Filter. Second, the optimal attitude in terms of $\delta\boldsymbol{\theta}$ may be found by taking the differential of Eq. 5.60 and setting the result equal to zero,

$$dJ(\delta\boldsymbol{\theta}) = [\delta\boldsymbol{\theta}^T \mathcal{F}_{\theta\theta}] d\boldsymbol{\theta} = 0 \quad (5.61)$$

Because $d\boldsymbol{\theta}$ is an independent differential it is arbitrary, and it's coefficient must be equal to zero if $dJ(\delta\boldsymbol{\theta})$ is to be zero. Further, because $\mathcal{F}_{\theta\theta}$ is full rank, it has no null space. Therefore, it is clear that the optimal solution is at $\delta\boldsymbol{\theta} = \mathbf{0}$. Because $\delta\boldsymbol{\theta}$ represents a rotation with respect to the estimated attitude, it is no surprise that the optimal solution should occur at $\delta\boldsymbol{\theta} = \mathbf{0}$. Further, if the second differential of Eq. 5.60 is greater than zero, then the optimal solution minimizes the objective function. Therefore, taking the second differential yields

$$d^2 J(\delta\boldsymbol{\theta}) = d\boldsymbol{\theta}^T \mathcal{F}_{\theta\theta} d\boldsymbol{\theta} > 0 \quad (5.62)$$

The Fisher information matrix is positive definite, confirming that the optimal attitude does indeed minimize the objective function.

5.5 Development of SOAR Filter

5.5.1 Maximum Likelihood State Estimation

The objective is to achieve a maximum likelihood estimate of the state vector at time t_k , \mathbf{x}_k , given an *a priori* estimate and some set of observations,

\mathbf{y} . Begin by recalling Bayes Theorem,

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \quad (5.63)$$

where $p(\mathbf{x}|\mathbf{y})$ is the probability density function (PDF) that describes the probability of observing the state vector \mathbf{x} conditioned on the knowledge of the observation \mathbf{y} .

In the present optimization problem, suppose the objective is to maximize the likelihood of the state estimate at time t_k conditioned on m new measurements at time t_k and an *a priori* estimate of the state (with the *a priori* estimate conditioned on r old measurements). Using Bayes Theorem, this may be mathematically expressed as

$$\begin{aligned} \text{Max } p(\mathbf{x}_k|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{r+m}) & \quad (5.64) \\ & = \frac{p(\mathbf{y}_{r+1}, \dots, \mathbf{y}_{r+m}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r)}{p(\mathbf{y}_{r+1}, \dots, \mathbf{y}_{r+m})} \end{aligned}$$

Proceed by assuming that each of the new observations are independent,

$$p(\mathbf{y}_{r+1}, \dots, \mathbf{y}_{r+m}|\mathbf{x}_k) = \prod_{i=r+1}^{r+m} p(\mathbf{y}_i|\mathbf{x}_k) \quad (5.65)$$

such that the optimization problem in Eq. 5.64 may be rewritten as¹

$$\text{Max } p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{r+m}) = \frac{p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r) \prod_{i=r+1}^{r+m} p(\mathbf{y}_i | \mathbf{x}_k)}{p(\mathbf{y}_{r+1}, \dots, \mathbf{y}_{r+m})} \quad (5.66)$$

This formulation, of course, makes no assumption about the actual distributions of the state vector or the measurements. Now, if the *a priori* estimate of the state vector at time t_k is assumed to have a Gaussian distribution, then

$$p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r) = C_{1,k} \exp \left\{ -\frac{1}{2} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{P}_k^-)^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \right\} \quad (5.67)$$

where $C_{1,k}$ is a scalar that makes the integral of $p(\cdot)$ equal to unity. This scalar is a function of the covariance matrix. Moving forward, assume that the covariance, and hence $C_{1,k}$, is independent of the state. Similarly, if the measurement errors are assumed to have a Gaussian distribution

$$p(\mathbf{y}_i | \mathbf{x}_k) = C_{2,i} \exp \left\{ -\frac{1}{2} (\mathbf{y}_i - h_i(\hat{\mathbf{x}}_k^-))^T \mathbf{R}_i^{-1} (\mathbf{y}_i - h_i(\hat{\mathbf{x}}_k^-)) \right\} \quad (5.68)$$

For the moment, assume the measurements are line-of-sight unit vector observations such that $\mathbf{y}_i = (\tilde{\mathbf{e}}_i)_B$ and $h_i(\hat{\mathbf{x}}_k^-) = \mathbf{T}_B^I (\mathbf{e}_i)_I$. Note that because $(\tilde{\mathbf{e}}_i)_B$ is a unit vector, the domain of \mathbf{y}_i is the unit sphere. A more detailed

¹To help explicitly connect the PDFs shown in Eq. 5.66 to the notation used elsewhere in this dissertation, consider the following relations. Assume that the state \mathbf{x} is an $n \times 1$ vector given by $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$

$$\hat{\mathbf{x}}_k^+ = E[\mathbf{x}_k^+] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mathbf{x}_k p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{r+m}) dx_1 dx_2 \dots dx_n$$

$$\hat{\mathbf{x}}_k^- = E[\mathbf{x}_k^-] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mathbf{x}_k p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r) dx_1 dx_2 \dots dx_n$$

discussion of the PDF for a unit vector measurement, including the complete expression for the scalar $C_{2,i}$, is given in [120]. Making these substitutions,

$$p(\mathbf{y}_i|\mathbf{x}_k) = C_{2,i} \exp \left\{ -\frac{1}{2} [(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I]^T \mathbf{R}_i^{-1} [(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I] \right\} \quad (5.69)$$

Substituting the pseudoinverse for \mathbf{R}^{-1} from Eq. 3.142 and recalling the identity from Eq. 3.129,

$$p(\mathbf{y}_i|\mathbf{x}_k) = C_{2,i} \exp \left\{ -\frac{1}{2} \frac{1}{\sigma_{\phi,i}^2} [(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I]^T [(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I] \right\} \quad (5.70)$$

Therefore, substituting the result of Eq. 5.70 into Eq 5.65,

$$\begin{aligned} p(\mathbf{y}_{r+1}, \dots, \mathbf{y}_{r+m}|\mathbf{x}_k) &= \prod_{i=r+1}^{r+m} p(\mathbf{y}_i|\mathbf{x}_k) \\ &= C_{3,k} \exp \left\{ -\frac{1}{2} \sum_{i=r+1}^{r+m} \frac{1}{\sigma_{\phi,i}^2} \|(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I\|^2 \right\} \end{aligned} \quad (5.71)$$

where $C_{3,k} = \prod_{i=r+1}^{r+m} C_{2,i}$. Note that the term in the exponent of Eq. 5.71 is equivalent to the traditional Wahba Problem objective function to first order if $w_i = 1/\sigma_{\phi,i}^2$. This demonstrates that solutions to the Wahba Problem produce a maximum likelihood estimate of the attitude if $w_i = 1/\sigma_{\phi,i}^2$.

Substituting Eq. 5.67 and Eq. 5.71 into Eq. 5.66, a maximum likeli-

hood estimate of the state exists at

$$\begin{aligned} \text{Max } p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{r+m}) &= \frac{C_{1,k} C_{3,k}}{p(\mathbf{y}_{r+1}, \dots, \mathbf{y}_{r+m})} & (5.72) \\ &\times \exp \left\{ -\frac{1}{2} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{P}_k^-)^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \right\} \\ &\times \exp \left\{ -\frac{1}{2} \sum_{i=r+1}^{r+m} \frac{1}{\sigma_{\phi,i}^2} \|(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I\|^2 \right\} \end{aligned}$$

Recognizing that the maximum of $p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{r+m})$ occurs at the same value of the state as the maximum of $\ln [p(\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{r+m})]$, a maximum likelihood estimate of the state may also be found by finding

$$\begin{aligned} \text{Max } \ln [C_{1,k}] + \ln [C_{3,k}] - \ln [p(\mathbf{y}_{r+1}, \dots, \mathbf{y}_{r+m})] & & (5.73) \\ &- \frac{1}{2} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{P}_k^-)^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \\ &- \frac{1}{2} \sum_{i=r+1}^{r+m} \frac{1}{\sigma_{\phi,i}^2} \|(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I\|^2 \end{aligned}$$

Because the first three terms do not depend on the estimate of the state, they may be dropped from the objective function in determining the maximum likelihood estimate. Dropping these terms and rewriting as a minimization problem yields the following form of the objective function

$$\begin{aligned} \text{Min } L &= \frac{1}{2} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{P}_k^-)^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) & (5.74) \\ &+ \frac{1}{2} \sum_{i=r+1}^{r+m} \frac{1}{\sigma_{\phi,i}^2} \|(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I\|^2 \end{aligned}$$

5.5.2 Discussion of SOAR Filter State Vector

The objective of the present filter is to estimate the attitude of a vehicle along with a number of other vehicle parameters. To express the attitude, it is desired to use a minimal degree representation (a number of different three-dimensional parameterizations would work here) instead of the constrained four-dimensional quaternion. Here, the attitude will be represented by the vector $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \mathbf{e}_\theta \theta \quad (5.75)$$

Where $\boldsymbol{\theta}$ is expressed in the body frame and represents the rotation from the current best estimate of the attitude to the true attitude. As was observed earlier, this means that $\hat{\boldsymbol{\theta}}^- \equiv \mathbf{0}$. The implications of this selection are critical to the proper treatment of attitude in the filter and will be discussed in detail in the following sections.

In addition to estimating the attitude, the SOAR Filter also estimates other vehicle parameters that are defined by the parameter vector $\boldsymbol{\beta}$. Therefore, the state vector for the SOAR Filter is given by

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\beta} \end{bmatrix} \quad (5.76)$$

5.5.3 Reformulation of SOAR Objective Function

Recalling the relation between the covariance matrix and the Fisher information matrix from Eq. 5.49, and define the components of the partitioned

Fisher information matrix as

$$\mathbf{P}^{-1} \approx \mathbf{F} = \begin{bmatrix} \mathbf{F}_{\theta\theta} & \mathbf{F}_{\theta\beta} \\ \mathbf{F}_{\beta\theta} & \mathbf{F}_{\beta\beta} \end{bmatrix} \quad (5.77)$$

Substituting this result at time k into the objective function from Eq. 5.74 yields

$$\begin{aligned} \text{Min } L = & \frac{1}{2} \delta \boldsymbol{\theta}_k^T \mathbf{F}_{\theta\theta,k}^- \delta \boldsymbol{\theta}_k + \frac{1}{2} \delta \boldsymbol{\theta}_k^T \mathbf{F}_{\theta\beta,k}^- \delta \boldsymbol{\beta}_k + \frac{1}{2} \delta \boldsymbol{\beta}_k^T \mathbf{F}_{\beta\theta,k}^- \delta \boldsymbol{\theta}_k \\ & + \frac{1}{2} \delta \boldsymbol{\beta}_k^T \mathbf{F}_{\beta\beta,k}^- \delta \boldsymbol{\beta}_k + \frac{1}{2} \sum_{i=r+1}^{r+m} \frac{1}{\sigma_{\phi,i}^2} \|(\tilde{\mathbf{e}}_i)_B - \mathbf{T}_B^I(\mathbf{e}_i)_I\|^2 \end{aligned} \quad (5.78)$$

The subscript k is temporarily dropped in the following derivation to abbreviate notation. Care must be taken here not to confuse the $\mathcal{F}_{\theta\theta}$ generated from the inverse of the 3×3 attitude covariance matrix in Eq. 5.54 with the $\mathbf{F}_{\theta\theta}$ described in Eq. 5.78 and Eq. 5.77. The $\mathbf{F}_{\theta\theta}$ in Eq. 5.78 and Eq. 5.77 is the upper 3×3 matrix of the inverse of the entire covariance matrix, which contains both attitude and non-attitude states. The different font is chosen to help keep this distinction clear.

Further, recognizing that $\mathbf{F}_{\theta\beta} = \mathbf{F}_{\beta\theta}^T$ and using the form of the Wahba Problem cost function from Eq. 5.31,

$$\text{Min } L = \frac{1}{2} \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\theta}^- \delta \boldsymbol{\theta} + \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\beta}^- \delta \boldsymbol{\beta} + \frac{1}{2} \delta \boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^- \delta \boldsymbol{\beta} + \lambda_0^m - \bar{\mathbf{q}}^T \mathbf{K}^m \bar{\mathbf{q}} \quad (5.79)$$

where \mathbf{K}^m is the measurement Davenport matrix constructed using only the new measurements. If the measurements are unit vectors, then \mathbf{K}^m is computed as in the regular Wahba Problem. If the measurements are quaternions (as is sometimes the case for commercially available star trackers that provide

a quaternion estimate instead of the raw measurements), then the measured attitude and measurement covariance matrix are used to create \mathbf{B}^m from Eq. 5.55. Now, \mathbf{B}^m may be used to compute \mathbf{K}^m from Eq. 5.32 to Eq. 5.35.

To obtain a more useful expression for Eq. 5.79, begin by looking at the first term on the right hand side. Recalling the following relation for the inverse of a partitioned matrix

$$\begin{aligned} & \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} [\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}]^{-1} & -[\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}]^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}[\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}]^{-1} & [\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}]^{-1} \end{bmatrix} \end{aligned} \quad (5.80)$$

it may be shown through a little algebra that

$$\mathbf{F}_{\theta\theta} = \mathbf{P}_{\theta\theta}^{-1} + \mathbf{F}_{\theta\beta}\mathbf{F}_{\beta\beta}^{-1}\mathbf{F}_{\beta\theta} = \mathcal{F}_{\theta\theta} + \mathbf{F}_{\theta\beta}\mathbf{F}_{\beta\beta}^{-1}\mathbf{F}_{\beta\theta} \quad (5.81)$$

Therefore, the first term in Eq. 5.79 may be rewritten as

$$\frac{1}{2}\delta\boldsymbol{\theta}^T\mathbf{F}_{\theta\theta}^{-1}\delta\boldsymbol{\theta} = \frac{1}{2}\delta\boldsymbol{\theta}^T\mathcal{F}_{\theta\theta}^{-1}\delta\boldsymbol{\theta} + \frac{1}{2}\delta\boldsymbol{\theta}^T\mathbf{F}_{\theta\beta}^{-1}(\mathbf{F}_{\beta\beta}^{-1})^{-1}\mathbf{F}_{\beta\theta}^{-1}\delta\boldsymbol{\theta} \quad (5.82)$$

Now, from Eq. 5.60, it is clear that

$$\frac{1}{2}\delta\boldsymbol{\theta}^T\mathcal{F}_{\theta\theta}^{-1}\delta\boldsymbol{\theta} = J(\delta\boldsymbol{\theta}) - J(\hat{\mathbf{q}}^-) = J(\bar{\mathbf{q}}) - J(\hat{\mathbf{q}}^-) \quad (5.83)$$

and substituting the result from Eq. 5.31 into this expression yields

$$\frac{1}{2}\delta\boldsymbol{\theta}^T\mathcal{F}_{\theta\theta}^{-1}\delta\boldsymbol{\theta} = \lambda_0^- - \bar{\mathbf{q}}^T\mathbf{K}^-\bar{\mathbf{q}} - J(\hat{\mathbf{q}}^-) \quad (5.84)$$

Therefore, substituting the result of Eq. 5.84 into Eq. 5.82, it may be shown that

$$\frac{1}{2}\delta\boldsymbol{\theta}^T\mathbf{F}_{\theta\theta}^{-1}\delta\boldsymbol{\theta} = \lambda_0^- - \bar{\mathbf{q}}^T\mathbf{K}^-\bar{\mathbf{q}} - J(\hat{\mathbf{q}}^-) + \frac{1}{2}\delta\boldsymbol{\theta}^T\mathbf{F}_{\theta\beta}^{-1}(\mathbf{F}_{\beta\beta}^{-1})^{-1}\mathbf{F}_{\beta\theta}^{-1}\delta\boldsymbol{\theta} \quad (5.85)$$

As a brief aside, the *a-priori* Davenport matrix, \mathbf{K}^- , may be found by using Eq. 5.32 to Eq. 5.35 and the *a-priori* attitude profile matrix, \mathbf{B}^- . The *a-priori* attitude and covariance may be used to find \mathbf{B}^- from Eq. 5.55,

$$\mathbf{B}^- = \left[\frac{1}{2} \text{tr} \left[(\mathbf{P}_{\theta\theta}^-)^{-1} \right] \mathbf{I}_{3 \times 3} - (\mathbf{P}_{\theta\theta}^-)^{-1} \right] \mathbf{T}_{\hat{\mathbf{q}}^-}^I \quad (5.86)$$

where $\mathbf{T}_{\hat{\mathbf{q}}^-}^I$ is found through Eq. 3.11 using $\hat{\mathbf{q}}^-$ as the attitude.

Substituting the result of Eq. 5.85 into the objective function from Eq. 5.79 and combining like terms,

$$\begin{aligned} \text{Min } L = & (\lambda_0^m + \lambda_0^-) - J(\hat{\mathbf{q}}^-) - \bar{\mathbf{q}}^T (\mathbf{K}^m + \mathbf{K}^-) \bar{\mathbf{q}} \\ & + \frac{1}{2} \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\theta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \delta \boldsymbol{\theta} + \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\beta}^- \delta \boldsymbol{\beta} + \frac{1}{2} \delta \boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^- \delta \boldsymbol{\beta} \end{aligned} \quad (5.87)$$

Note that this expression contains both $\bar{\mathbf{q}}$ and $\delta \boldsymbol{\theta}$. These are simply two different ways of expressing the attitude. Therefore, in order to simplify this expression, note the following relations associated with rotations and quaternion multiplication

$$\begin{bmatrix} \delta \boldsymbol{\theta} / 2 \\ 1 \end{bmatrix} \approx \delta \bar{\mathbf{q}} = \bar{\mathbf{q}} \otimes (\hat{\mathbf{q}}^-)^{-1} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \otimes \begin{bmatrix} -\hat{\mathbf{q}}^- \\ \hat{q}_4^- \end{bmatrix} \quad (5.88)$$

Using the definition of the quaternion product, proceed by showing that

$$\begin{aligned} \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \otimes \begin{bmatrix} -\hat{\mathbf{q}}^- \\ \hat{q}_4^- \end{bmatrix} &= \begin{bmatrix} \hat{q}_4^- \mathbf{q} - q_4 \hat{\mathbf{q}}^- + \mathbf{q} \times \hat{\mathbf{q}}^- \\ q_4 \hat{q}_4^- + \mathbf{q}^T \hat{\mathbf{q}}^- \end{bmatrix} \\ &= \begin{bmatrix} \hat{q}_4^- \mathbf{I}_{3 \times 3} - [\hat{\mathbf{q}}^- \times] & -\hat{\mathbf{q}}^- \\ (\hat{\mathbf{q}}^-)^T & \hat{q}_4^- \end{bmatrix} \bar{\mathbf{q}} \end{aligned} \quad (5.89)$$

Therefore, define the 3×4 matrix $\boldsymbol{\Psi}(\hat{\mathbf{q}}^-)$ as

$$\boldsymbol{\Psi}(\hat{\mathbf{q}}^-) = \begin{bmatrix} \hat{q}_4^- \mathbf{I}_{3 \times 3} - [\hat{\mathbf{q}}^- \times] & -\hat{\mathbf{q}}^- \end{bmatrix} \quad (5.90)$$

such that

$$\delta\boldsymbol{\theta} = 2\boldsymbol{\Psi}(\hat{\mathbf{q}}^-) \bar{\mathbf{q}} \quad (5.91)$$

Now, substituting this identity into Eq. 5.87 allows the objective function to be written as

$$\begin{aligned} \text{Min } L = & (\lambda_0^m + \lambda_0^-) - J(\hat{\mathbf{q}}^-) - \bar{\mathbf{q}}^T (\mathbf{K}^m + \mathbf{K}^-) \bar{\mathbf{q}} \\ & + 2\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi}(\hat{\mathbf{q}}^-) \bar{\mathbf{q}} \\ & + 2\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- \delta\boldsymbol{\beta} + \frac{1}{2} \delta\boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^- \delta\boldsymbol{\beta} \end{aligned} \quad (5.92)$$

5.5.4 Finding the Optimal State Update in SOAR

Using a similar approach as for the optimal solution for the regular Wahba problem, adjoin the attitude quaternion unity norm constraint to the objective function using a Lagrange multiplier, λ , such that

$$\begin{aligned} \text{Min } \mathcal{L}(\bar{\mathbf{q}}, \boldsymbol{\beta}, \lambda) = & (\lambda_0^m + \lambda_0^-) - J(\hat{\mathbf{q}}^-) - \bar{\mathbf{q}}^T (\mathbf{K}^m + \mathbf{K}^-) \bar{\mathbf{q}} \\ & + 2\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi}(\hat{\mathbf{q}}^-) \bar{\mathbf{q}} \\ & + 2\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- \delta\boldsymbol{\beta} + \frac{1}{2} \delta\boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^- \delta\boldsymbol{\beta} + \lambda (\bar{\mathbf{q}}^T \bar{\mathbf{q}} - 1) \end{aligned} \quad (5.93)$$

Taking the differential of $\mathcal{L}(\bar{\mathbf{q}}, \boldsymbol{\beta}, \lambda)$,

$$\begin{aligned} d\mathcal{L}(\bar{\mathbf{q}}, \boldsymbol{\beta}, \lambda) = & \{-2\bar{\mathbf{q}}^T (\mathbf{K}^m + \mathbf{K}^-) \\ & + 4\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi}(\hat{\mathbf{q}}^-) \\ & + 2\delta\boldsymbol{\beta}^T \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi}(\hat{\mathbf{q}}^-) + 2\lambda\bar{\mathbf{q}}^T\} d\bar{\mathbf{q}} \\ & + \{2\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- + \delta\boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^-\} d\boldsymbol{\beta} + \{\bar{\mathbf{q}}^T \bar{\mathbf{q}} - 1\} d\lambda = 0 \end{aligned} \quad (5.94)$$

where the first three components of $d\bar{\mathbf{q}}$ are independent differentials and the fourth component is a dependent differential (this distinction is required because of the quaternion unity norm constraint). The differentials have been combined back into $d\bar{\mathbf{q}}$ for brevity. Setting the coefficient of $d\lambda$ to zero produces

$$\bar{\mathbf{q}}^T \bar{\mathbf{q}} - 1 = 0 \quad (5.95)$$

which simply states that the quaternion unity norm constraint must be satisfied. Assuming no constraints on the non-attitude states, $d\boldsymbol{\beta}$ is an independent differential. Therefore, if $d\mathcal{J}(\bar{\mathbf{q}}, \boldsymbol{\beta}, \lambda)$ is to be zero for an arbitrary $d\boldsymbol{\beta}$, the coefficient of $d\boldsymbol{\beta}$ must be zero,

$$2\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- + \delta\boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^- = 0 \quad (5.96)$$

Rearranging and solving for $\delta\boldsymbol{\beta}$ (again recall that $\mathbf{F}_{\beta\beta}$ is symmetric),

$$\delta\boldsymbol{\beta} = -2(\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi}(\hat{\mathbf{q}}^-) \bar{\mathbf{q}} \quad (5.97)$$

Now move to the term involving $d\bar{\mathbf{q}} = [d\mathbf{q}^T dq_4]^T$ in Eq. 5.94. Because $d\mathbf{q}$ is an independent differential, its coefficients must be zero. Additionally, the Lagrange multiplier λ is picked such that the coefficient of the dependent differential dq_4 is also zero. Therefore, the coefficient of $d\bar{\mathbf{q}}$ in Eq. 5.94 must be zero,

$$\begin{aligned} & -2\bar{\mathbf{q}}^T (\mathbf{K}^m + \mathbf{K}^-) + 4\bar{\mathbf{q}}^T \boldsymbol{\Psi}(\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi}(\hat{\mathbf{q}}^-) \\ & + 2\delta\boldsymbol{\beta}^T \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi}(\hat{\mathbf{q}}^-) + 2\lambda\bar{\mathbf{q}}^T = 0 \end{aligned} \quad (5.98)$$

Substituting the optimal value of $\delta\boldsymbol{\beta}$ from Eq. 5.97,

$$\begin{aligned} -\bar{\mathbf{q}}^T (\mathbf{K}^m + \mathbf{K}^-) + 2\bar{\mathbf{q}}^T \boldsymbol{\Psi} (\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi} (\hat{\mathbf{q}}^-) \\ - 2\bar{\mathbf{q}}^T \boldsymbol{\Psi} (\hat{\mathbf{q}}^-)^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi} (\hat{\mathbf{q}}^-) + \lambda\bar{\mathbf{q}}^T = 0 \end{aligned} \quad (5.99)$$

The middle two terms cancel, leaving

$$-\bar{\mathbf{q}}^T (\mathbf{K}^m + \mathbf{K}^-) + \lambda\bar{\mathbf{q}}^T = 0 \quad (5.100)$$

Therefore, if \mathbf{K}^+ is given by

$$\mathbf{K}^+ = \mathbf{K}^m + \mathbf{K}^- \quad (5.101)$$

then Eq. 5.100 may be rewritten as

$$-\bar{\mathbf{q}}^T \mathbf{K}^+ + \lambda\bar{\mathbf{q}}^T = 0 \quad (5.102)$$

Recognizing that \mathbf{K}^+ is symmetric and that the solution to Eq. 5.102 is the *a-posteriori* attitude estimate, which allows $\bar{\mathbf{q}}$ to be replaced by $\hat{\mathbf{q}}^+$,

$$\mathbf{K}^+ \hat{\mathbf{q}}^+ = \lambda \hat{\mathbf{q}}^+ \quad (5.103)$$

The optimal $\hat{\mathbf{q}}^+$ may be found using any solution method to the normal Wahba Problem. Once $\hat{\mathbf{q}}^+$ is known, $\delta\boldsymbol{\beta}$ may be found using Eq. 5.97. The updated parameter vector, therefore, is simply given by

$$\boldsymbol{\beta}^+ = \boldsymbol{\beta}^- + \delta\boldsymbol{\beta} = \boldsymbol{\beta}^- - 2 (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \boldsymbol{\Psi} (\hat{\mathbf{q}}^-) \hat{\mathbf{q}}^+ \quad (5.104)$$

5.5.5 Finding the Optimal Covariance Update in SOAR

Now that the optimal state update has been derived, it is necessary to compute the corresponding update to the state covariance matrix. Begin by rewriting the objective function from Eq. 5.87 as

$$\begin{aligned} \text{Min } L = & (\lambda_0^m + \lambda_0^-) - J(\hat{\mathbf{q}}^-) - \text{tr} \left[\mathbf{T}_B^I (\mathbf{B}^m + \mathbf{B}^-)^T \right] \\ & + \frac{1}{2} \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \delta \boldsymbol{\theta} + \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\beta}^- \delta \boldsymbol{\beta} + \frac{1}{2} \delta \boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^- \delta \boldsymbol{\beta} \end{aligned} \quad (5.105)$$

As was done in finding the covariance for the regular Wahba Problem in Eq. 5.52, rewrite the attitude term in terms of $\delta \boldsymbol{\theta}$, where $\delta \boldsymbol{\theta}$ is a small angle rotation from the best estimate of the body frame,

$$\begin{aligned} \text{Min } L = & (\lambda_0^m + \lambda_0^-) - J(\hat{\mathbf{q}}^-) \\ & - \text{tr} \left[\left(\mathbf{I}_{3 \times 3} + [-\delta \boldsymbol{\theta} \times] + \frac{1}{2} [-\delta \boldsymbol{\theta} \times]^2 \right) \mathbf{T}_{\hat{B}}^I (\mathbf{B}^m + \mathbf{B}^-)^T \right] \\ & + \frac{1}{2} \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \delta \boldsymbol{\theta} + \delta \boldsymbol{\theta}^T \mathbf{F}_{\theta\beta}^- \delta \boldsymbol{\beta} + \frac{1}{2} \delta \boldsymbol{\beta}^T \mathbf{F}_{\beta\beta}^- \delta \boldsymbol{\beta} \end{aligned} \quad (5.106)$$

If the relation in Eq. 5.101 is true, then the following must also be true

$$\mathbf{B}^+ = \mathbf{B}^m + \mathbf{B}^- \quad (5.107)$$

Substituting this result into Eq. 5.106, the *a-posteriori* Fisher information matrix may be computed using the relation given in Eq. 5.47. Therefore, the differentiation of Eq. 5.106 will yield the following results

$$\mathbf{F}_{\theta\theta}^+ = \text{tr} \left[\mathbf{T}_{\hat{B}}^I (\mathbf{B}^+)^T \right] \mathbf{I}_{3 \times 3} - \mathbf{T}_{\hat{B}}^I (\mathbf{B}^+)^T + \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \quad (5.108)$$

$$\mathbf{F}_{\theta\beta}^+ = (\mathbf{F}_{\beta\theta}^+)^T = \mathbf{F}_{\theta\beta}^- \quad (5.109)$$

$$\mathbf{F}_{\beta\beta}^+ = \mathbf{F}_{\beta\beta}^- \quad (5.110)$$

This makes sense in that the only new information available is in the attitude estimate. The present implementation of the SOAR Filter only allows line-of-sight unit vector measurements and attitude measurements, therefore improved estimates of the parameter vector $\boldsymbol{\beta}$ is only possible through knowledge of the dynamics and correlation between the estimates of the attitude and parameters. Further, it is interesting to note the relationship between Eq. 5.108 and Eq. 5.54. The first two terms in Eq. 5.108 are what would be expected from Eq. 5.54 to go from \mathbf{B}^+ to $\mathbf{F}_{\theta\theta}^+$ in a system where only attitude states are estimated. The third term, which is new, follows from the inclusion of non-attitude states.

To compute the *a-posteriori* covariance matrix, recall the relation for the inverse of a segmented matrix given in Eq. 5.80. From this relation, it is straightforward to see that

$$\mathbf{P}_{\theta\theta}^+ = \left[\mathbf{F}_{\theta\theta}^+ - \mathbf{F}_{\theta\beta}^+ (\mathbf{F}_{\beta\beta}^+)^{-1} \mathbf{F}_{\beta\theta}^+ \right]^{-1} \quad (5.111)$$

$$\mathbf{P}_{\beta\beta}^+ = \left[\mathbf{F}_{\beta\beta}^+ - \mathbf{F}_{\beta\theta}^+ (\mathbf{F}_{\theta\theta}^+)^{-1} \mathbf{F}_{\theta\beta}^+ \right]^{-1} \quad (5.112)$$

$$\mathbf{P}_{\beta\theta}^+ = (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \mathbf{P}_{\theta\theta}^+ \quad (5.113)$$

A useful identity may be shown by expanding these three equations using the Woodbury identity.² As an example, the this identity allows the expression

²The Woodbury identity for matrix inverses is given by: $[\mathbf{A} + \mathbf{C} \mathbf{B} \mathbf{C}^T]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{C} [\mathbf{B}^{-1} + \mathbf{C}^T \mathbf{A}^{-1} \mathbf{C}]^{-1} \mathbf{C}^T \mathbf{A}^{-1}$

for $\mathbf{P}_{\beta\beta}^+$ to be rewritten as

$$\mathbf{P}_{\beta\beta}^+ = (\mathbf{F}_{\beta\beta}^+)^{-1} + (\mathbf{F}_{\beta\beta}^+)^{-1} \mathbf{F}_{\beta\theta}^+ \mathbf{P}_{\theta\theta}^+ \mathbf{F}_{\theta\beta}^+ (\mathbf{F}_{\beta\beta}^+)^{-1} \quad (5.114)$$

Recalling the relations of Eq. 5.109 and Eq. 5.110,

$$\mathbf{P}_{\beta\beta}^+ = \mathbf{P}_{\beta\beta}^- + (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- [\mathbf{P}_{\theta\theta}^+ - \mathbf{P}_{\theta\theta}^-] \mathbf{F}_{\theta\beta}^- (\mathbf{F}_{\beta\beta}^-)^{-1} \quad (5.115)$$

Using the same procedure, a similar relation may be shown for $\mathbf{P}_{\theta\theta}^+$

$$\mathbf{P}_{\theta\theta}^+ = \mathbf{P}_{\theta\theta}^- + (\mathbf{F}_{\theta\theta}^-)^{-1} \mathbf{F}_{\theta\beta}^- [\mathbf{P}_{\beta\beta}^+ - \mathbf{P}_{\beta\beta}^-] \mathbf{F}_{\beta\theta}^- (\mathbf{F}_{\theta\theta}^-)^{-1} \quad (5.116)$$

The off-diagonal elements may be found using a similar approach. Directly from Eq. 5.80 one may write

$$\mathbf{P}_{\beta\theta}^+ = (\mathbf{F}_{\beta\beta}^-)^{-1} \mathbf{F}_{\beta\theta}^- \mathbf{P}_{\theta\theta}^+ \quad (5.117)$$

The same results may also be found using the standard definition of the covariance matrix (i.e. a covariance/uncertainty approach rather than an information approach). Begin by considering the definition of the *a-posteriori* attitude covariance

$$\mathbf{P}_{\theta\theta}^+ = E \left[(\boldsymbol{\theta}^+ - E[\boldsymbol{\theta}]) (\boldsymbol{\theta}^+ - E[\boldsymbol{\theta}])^T \right] \quad (5.118)$$

Rewriting in terms of the *a-priori* attitude and the attitude update, $\boldsymbol{\theta}^+ = \boldsymbol{\theta}^- + \Delta\boldsymbol{\theta}$, and assuming an unbiased estimator, $E[\Delta\boldsymbol{\theta}] = \mathbf{0}$,

$$\mathbf{P}_{\theta\theta}^+ = E \left[(\boldsymbol{\theta}^- + \Delta\boldsymbol{\theta} - E[\boldsymbol{\theta}]) (\boldsymbol{\theta}^- + \Delta\boldsymbol{\theta} - E[\boldsymbol{\theta}])^T \right] \quad (5.119)$$

$$\mathbf{P}_{\theta\theta}^+ = \mathbf{P}_{\theta\theta}^- + E[\Delta\boldsymbol{\theta} \Delta\boldsymbol{\theta}^T] \quad (5.120)$$

Turning attention to the *a-posteriori* covariance for the parameter vector,

$$\mathbf{P}_{\beta\beta}^+ = E \left[(\boldsymbol{\beta}^+ - E[\boldsymbol{\beta}]) (\boldsymbol{\beta}^+ - E[\boldsymbol{\beta}])^T \right] \quad (5.121)$$

where it is assumed that $E[\Delta\boldsymbol{\beta}] = \mathbf{0}$. As with $\mathbf{P}_{\theta\theta}^+$ this yields

$$\mathbf{P}_{\beta\beta}^+ = \mathbf{P}_{\beta\beta}^- + E[\Delta\boldsymbol{\beta} \Delta\boldsymbol{\beta}^T] \quad (5.122)$$

Using the parameter vector update from Eq. 5.97,

$$\mathbf{P}_{\beta\beta}^+ = \mathbf{P}_{\beta\beta}^- + E \left[(-\mathbf{F}_{\beta\beta}^{-1} \mathbf{F}_{\beta\theta} \Delta\boldsymbol{\theta}) (-\mathbf{F}_{\beta\beta}^{-1} \mathbf{F}_{\beta\theta} \Delta\boldsymbol{\theta})^T \right] \quad (5.123)$$

Recognizing that the only random variable is $\Delta\boldsymbol{\theta}$, this may be rewritten as

$$\mathbf{P}_{\beta\beta}^+ = \mathbf{P}_{\beta\beta}^- + \mathbf{F}_{\beta\beta}^{-1} \mathbf{F}_{\beta\theta} E[\Delta\boldsymbol{\theta} \Delta\boldsymbol{\theta}^T] \mathbf{F}_{\theta\beta} \mathbf{F}_{\beta\beta}^{-1} \quad (5.124)$$

Substituting for $E[\Delta\boldsymbol{\theta} \Delta\boldsymbol{\theta}^T]$ from Eq. 5.120 produces the same equation for $\mathbf{P}_{\beta\beta}^+$ as obtained through inversion of the Fisher information matrix in Eq. 5.115. A similar procedure will verify the expressions for $\mathbf{P}_{\theta\theta}^+$ and $\mathbf{P}_{\beta\theta}^+$. The details of this equivalence for $\mathbf{P}_{\theta\theta}^+$ and $\mathbf{P}_{\beta\theta}^+$ are not presented here for brevity.

5.5.6 Propagation of State and Covariance for SOAR

The only remaining step is to propagate the *a-posteriori* state estimate and covariance at time t_k forward in time to create the *a-priori* state estimate and covariance at time t_{k+1} . Consider a system with the following nonlinear dynamic model:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(t)\mathbf{w}(t) \quad (5.125)$$

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) \quad (5.126)$$

Proceed by taking the Taylor series expansion of $\mathbf{f}(\mathbf{x}, \mathbf{u}, t)$ about the estimated trajectory and keep only first-order terms,

$$\dot{\mathbf{x}} = \left[\mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t) + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}(t)} \right) (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \right] + \mathbf{G}(t)\mathbf{w}(t) \quad (5.127)$$

Defining the matrix $\mathbf{F}(\hat{\mathbf{x}}(t), t)$ as

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}(t)} \quad (5.128)$$

and establishing the following state error vectors,

$$\boldsymbol{\delta}\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}} \quad \boldsymbol{\delta}\dot{\mathbf{x}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} \quad (5.129)$$

allows for Eq. 5.127 to be rewritten as the following linearized model:

$$\boldsymbol{\delta}\dot{\mathbf{x}} = \mathbf{F}(\hat{\mathbf{x}}, t) \boldsymbol{\delta}\mathbf{x} + \mathbf{G}(t)\mathbf{w}(t) \quad (5.130)$$

This is a standard result that seen in the implementation of the EKF; it is well-known and discussed thoroughly in the literature.[81, 151, 152] Such a linear model is known to admit a solution of the form

$$\boldsymbol{\delta}\mathbf{x}(t) = \boldsymbol{\Phi}(t, t_0) \boldsymbol{\delta}\mathbf{x}(t_0) + \int_{t_0}^t \boldsymbol{\Phi}(t, \tau) \mathbf{G}(\tau)\mathbf{w}(\tau) d\tau \quad (5.131)$$

where $\boldsymbol{\Phi}(t, t_0)$ is the state transition matrix from t_0 to t . Therefore, assuming that the state error and the process noise are uncorrelated, i.e. $E[\boldsymbol{\delta}\mathbf{x}(t)\mathbf{w}(t)^T] = 0$, it may be shown that

$$\begin{aligned} \mathbf{P} &= E[\boldsymbol{\delta}\mathbf{x}\boldsymbol{\delta}\mathbf{x}^T] \\ &= \boldsymbol{\Phi}(t, t_0) \mathbf{P}_0 \boldsymbol{\Phi}(t, t_0)^T + \int_{t_0}^t \boldsymbol{\Phi}(t, \tau) \mathbf{G}(\tau) \mathbf{Q}(\tau) \mathbf{G}^T \boldsymbol{\Phi}(t, \tau)^T d\tau \end{aligned} \quad (5.132)$$

where

$$\mathbf{Q}(\tau) = E [\mathbf{w}(\tau)\mathbf{w}(\tau)^T] \quad (5.133)$$

Taking the time derivative of Eq. 5.132 (which requires the application of Leibniz's Rule) and recognizing that $\dot{\Phi} = \mathbf{F}\Phi$, yields the well known equation for the propagation of the covariance matrix typically seen in the discrete-continuous EKF:

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}, t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(\hat{\mathbf{x}}, t)^T + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}(t)^T \quad (5.134)$$

This expression is used to propagate the covariance between measurement updates in SOAR.

5.6 Construction of SOAR Algorithm

All the theoretical components required to create the SOAR Filter are now in place. A brief summary of the SOAR Filter is provided in the following paragraphs. A flow chart of the SOAR Filter is shown in Fig. 5.1.

Given an *a-priori* attitude estimate and covariance matrix, compute the *a-priori* attitude profile matrix from Eq. 5.55. Use this to create the *a-priori* Davenport matrix using Eq. 5.32 through Eq. 5.35.

Next, compute the measurement attitude profile matrix. If the measurements are from unit vectors, create the measurement attitude profile matrix using Eq. 5.15. If, on the other hand, a measured attitude is available, compute the measurement attitude profile matrix using Eq. 5.55. Because the Davenport matrix \mathbf{K} is a linear homogeneous function of \mathbf{B} , the measurement

attitude profile matrices for all measurement sources (whether they come from unit vector measurements or quaternion measurements) may be added before using Eq. 5.32 through Eq. 5.35 to get the measurement Davenport matrix at any given epoch.

The SOAR Davenport matrix may now be computed through Eq. 5.101. Use any solution method to the Wahba Problem to solve for the optimal attitude. With the optimal attitude found, use Eq. 5.97 to compute the update to the parameter vector β . Finally compute the updated covariance matrix by taking the inverse of the Fisher information matrix found using Eq. 5.108 through Eq. 5.110.

To state vector may be propagated to the next measurement epoch using Eq. 5.126, and the covariance matrix may be propagated using Eq. 5.134. Once at the next epoch, the procedure is repeated.

5.7 Comparison of SOAR with Other Attitude Filters

5.7.1 Comparison of the SOAR Filter with the Multiplicative Extended Kalman Filter

The central idea behind the MEKF is to use the unconstrained three-parameter representation $\delta\theta$ instead of the constrained four-parameter attitude quaternion in the state vector of a regular EKF.[169,176] This means that the MEKF estimates $\delta\hat{\theta}$ and then updates the attitude quaternion by

$$\bar{\mathbf{q}}^+ \approx \begin{bmatrix} \delta\hat{\theta}/2 \\ 1 \end{bmatrix} \otimes \bar{\mathbf{q}}^- \quad (5.135)$$

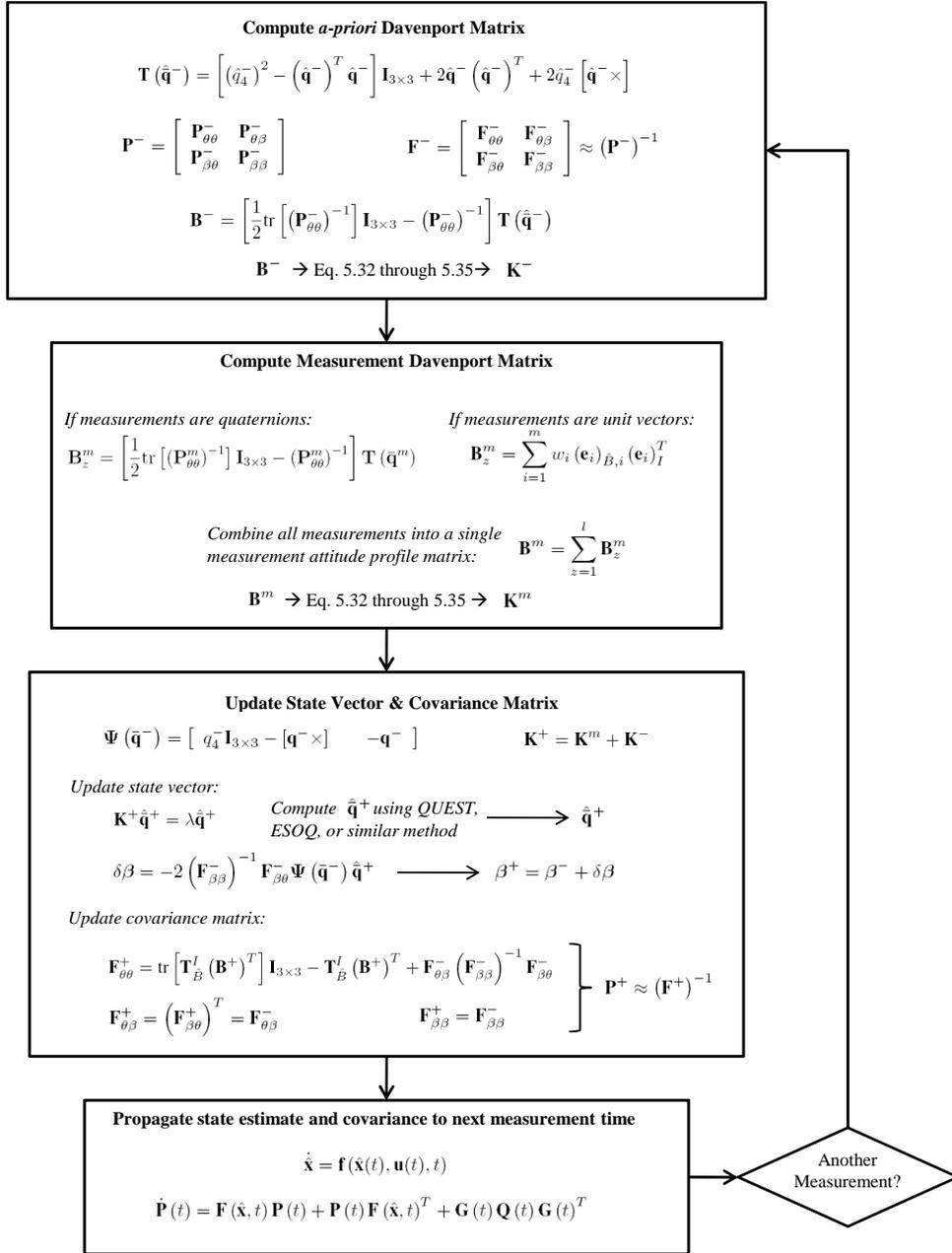


Figure 5.1: SOAR Filter flow chart.

This update only maintains the quaternion unity norm constraint to first order, making it common practice to normalize the *a-posteriori* quaternion, $\bar{\mathbf{q}}^+ = \bar{\mathbf{q}}^+ / \|\bar{\mathbf{q}}^+\|$.

An important similarity between the MEKF and the SOAR Filter is the objective function used to find the optimal update. Both the MEKF and SOAR Filter explicitly find the minimum variance state update (under the assumption that errors are Gaussian) and therefore minimize $J = \text{tr} [\mathbf{P}^+]$. Therefore, the MEKF and the SOAR Filter are expected to behave in a similar fashion when the errors are small and the MEKF linearization assumptions are good. Simulation confirms that this is the case. A second important similarity is that both the MEKF and SOAR can estimate non-attitude states.

Unlike the MEKF, the SOAR Filter deals directly with the nonlinear nature of attitude representations and the Wahba Problem. Rather than linearizing the update equation as is required in the MEKF, the SOAR Filter explicitly solves the constrained quadratic cost function. This allows the SOAR Filter to directly compute the attitude quaternion.

5.7.2 Comparison of SOAR with Filter QUEST and REQUEST

Filter QUEST[161, 162] and REQUEST[163] are both based on sequentialization of the Wahba Problem. Performing this sequentialization produces the following recursive filter equation (using the same nomenclature as for the SOAR Filter) that is called “Filter QUEST” in the literature,[161]

$$\mathbf{B}_{k+1}^+ = \mathbf{B}_{k+1}^m + \alpha_{k+1} \Phi(t_{k+1}, t_k) \mathbf{B}_k^+ \quad (5.136)$$

where $\alpha_{k+1} \in [0 \ 1]$ and represents reduction in knowledge associated with the propagation. This is a classic fading memory filter [152] where $\alpha_{k+1} = 0$ causes the filter to ignore the old measurements and $\alpha_{k+1} = 1$ makes the filter think there is no process noise and the measurements are combined as if they were all taken simultaneously. Therefore, heuristically speaking, the fading memory factor α_{k+1} should look something like $\sum \sigma(t_k)_{\phi,i}^2 / \sigma(t_{k+1})_{\phi,i}^2$. Note, however, that the Filter QUEST algorithm picks a suboptimal value for α_{k+1} to reduce complexity and increase algorithm speed.

Because the Davenport matrix \mathbf{K} is a linear homogeneous function of attitude profile matrix \mathbf{B} , Eq. 5.136 may be written in terms of \mathbf{K} instead of \mathbf{B} . This represents the fundamental difference between the Filter QUEST and REQUEST algorithm: Filter QUEST is a sequentialization on \mathbf{B} and REQUEST is a sequentialization on \mathbf{K} . Shuster provides an excellent discussion of the relation between these two algorithms,[164] which are shown to be equivalent when the same selection is made for α_{k+1} . Therefore, the conclusions made in the comparison with Filter QUEST also hold for REQUEST.

Although not generally true, it is possible to analytically select the forgetfulness factor (α_{k+1} in Eq. 5.136) under specific scenarios. Consider a system with the following assumptions:

1. Only three unit vector measurements are available
2. Each unit vector measurement lies along one of the coordinate axes with a measurement standard deviation of σ_ϕ

3. The process noise is a zero mean white noise process with $\mathbf{Q} = q\mathbf{I}_{3 \times 3}$
4. Steady state filter covariances
5. The attitude covariance matrix is diagonal, $\mathbf{P}_{\theta\theta} = p\mathbf{I}_{3 \times 3}$

Shuster demonstrates[161, 162] that under these conditions the optimal α_{k+1} may be analytically computed as

$$\alpha_{k+1} = \frac{\sigma_\phi^2/q + 1 - \sqrt{1 + 2\sigma_\phi^2/q}}{\sigma_\phi^2/q} \quad (5.137)$$

This simple equation for α_{k+1} provides an excellent approximation of the more general optimal fading memory factor. Note, however, that a typical star tracker may track many more than three stars and all the unit vector observations will be separated by an angle less than the camera FOV (i.e. it is not possible to obtain measurements along each of the coordinate axes). This means that the α_{k+1} given in Eq. 5.137 is suboptimal in the general sense.

The SOAR Filter removes all five of the assumptions made in computing α_{k+1} . As discussed above, the SOAR Filter provides the optimal state update under the assumption that state vector error and measurement errors are normally distributed. The SOAR Filter is also capable of estimating other states in addition to attitude (recall that one of the major drawbacks to Filter QUEST and REQUEST is their inability to also estimate non-attitude states). These advantages, however, come at the expense of additional computations.

5.7.3 Comparison of SOAR with Optimal-REQUEST

The Optimal-REQUEST algorithm[165] finds the attitude that minimizes the objective function

$$\text{Min } J_{Opt-RQ} = \text{tr} \left[E \left[\Delta \mathbf{K} (\Delta \mathbf{K})^T \right] \right] \quad (5.138)$$

where $\mathbf{K} = \hat{\mathbf{K}} + \Delta \mathbf{K}$. Recognizing that $\Delta \mathbf{K}$ is symmetric, it is straightforward to show that the 2-norm and Frobenius norm of the matrix $\Delta \mathbf{K}$ are given by

$$\|\Delta \mathbf{K}\|_2^2 = \lambda_{max} \left(\Delta \mathbf{K} (\Delta \mathbf{K})^T \right) = \lambda_{max}^2 (\Delta \mathbf{K}) \quad (5.139)$$

$$\|\Delta \mathbf{K}\|_F^2 = \text{tr} \left[\Delta \mathbf{K} (\Delta \mathbf{K})^T \right] \quad (5.140)$$

This means that Optimal-REQUEST minimizes the expected value of the square of the Frobenius norm of $\Delta \mathbf{K}$.

Recalling an important result from eigenvalue-eigenvector stability,[177] the estimate of the attitude quaternion may be written as,

$$\hat{\mathbf{q}} = \bar{\mathbf{q}} + \sum_{j=2}^4 \left[\frac{\mathbf{u}_j^T \Delta \mathbf{K} \bar{\mathbf{q}}}{\lambda_1 - \lambda_j} \right] \mathbf{u}_j + H.O.T. \quad (5.141)$$

where \mathbf{u}_j are the eigenvectors and $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$ are the eigenvalues of the true Davenport matrix, \mathbf{K} . Also recall that the true quaternion is associated with the largest eigenvalue of \mathbf{K} (i.e. $\mathbf{u}_1 \equiv \bar{\mathbf{q}}$). Because all \mathbf{u}_j are assumed to have unity norm and $\Delta \mathbf{K}$ must be a symmetric matrix, it is straightforward to show (Jordan decomposition) that $\mathbf{u}_j^T \Delta \mathbf{K} \bar{\mathbf{q}} \leq \lambda_{max} (\Delta \mathbf{K})$. Because it can also be shown that $\text{tr} [\mathbf{P}_{\theta\theta}] = 4 \text{tr} \left[E \left[(\hat{\mathbf{q}} - \bar{\mathbf{q}}) (\hat{\mathbf{q}} - \bar{\mathbf{q}})^T \right] \right]$ to first order, then

$$\text{tr} [\mathbf{P}_{\theta\theta}] \leq 4E \left[\lambda_{max}^2 (\Delta \mathbf{K}) \right] \sum_{j=2}^4 \sum_{l=2}^4 \left[\frac{1}{(\lambda_1 - \lambda_j) (\lambda_1 - \lambda_l)} \right] \text{tr} [\mathbf{u}_j \mathbf{u}_l^T] \quad (5.142)$$

Note that the double summation is only a function of the true attitude and, therefore, may be moved outside of the expected value operator. Additionally, using the 2-norm identity from Eq. 5.139, the above may be rewritten as

$$\text{tr} [\mathbf{P}_{\theta\theta}] \leq 4E [\|\Delta\mathbf{K}\|_2^2] \zeta(\mathbf{K}) \quad (5.143)$$

where

$$\zeta(\mathbf{K}_k) = \sum_{j=2}^4 \sum_{l=2}^4 \left[\frac{1}{(\lambda_1 - \lambda_j)(\lambda_1 - \lambda_l)} \right] \text{tr} [\mathbf{u}_j \mathbf{u}_l^T] \quad (5.144)$$

Now, recalling that the 2-norm is always less than or equal to the Frobenius norm, $\|\Delta\mathbf{K}\|_2 \leq \|\Delta\mathbf{K}\|_F$, the above inequality may be extended as

$$\text{tr} [\mathbf{P}_{\theta\theta}] \leq 4E [\|\Delta\mathbf{K}\|_2^2] \zeta(\mathbf{K}) \leq 4E [\|\Delta\mathbf{K}\|_F^2] \zeta(\mathbf{K}) \quad (5.145)$$

The SOAR Filter, because it provides a maximum likelihood estimate and assumes normally distributed random variables, provides a minimum variance estimate of the attitude, i.e. the SOAR Filter minimizes $J_{SOAR} = \text{tr} [\mathbf{P}_{\theta\theta}]$. Now, recalling that $J_{Opt-RQ} = E [\|\Delta\mathbf{K}\|_F^2]$, will show that

$$J_{SOAR} \leq 4J_{Opt-RQ} \zeta(\mathbf{K}) \quad (5.146)$$

Therefore, Optimal-REQUEST and the SOAR Filter provide the minimum to two different objective functions. The objective functions of the SOAR Filter and Optimal-REQUEST are related by Eq. 5.146. It is important to note that the inequality of Eq. 5.146 should not be taken to suggest that the SOAR Filter's objective function is better - this inequality simply shows the relation between the two different objective functions. That being said, the SOAR

Filter has the advantage that it explicitly addresses the objective of providing a maximum likelihood estimate of the spacecraft attitude. Finally, because Optimal-REQUEST is an extension of the REQUEST algorithm (with optimal α_{k+1} chosen to minimize Eq. 5.138), the Optimal-REQUEST algorithm is also unable to estimate non-attitude states.

5.7.4 Comparison of the SOAR Filter with Extended-QUEST

The Extended-QUEST algorithm is the closest of the existing attitude filtering methods to the SOAR Filter. The Extended-QUEST algorithm is a solution to the following optimization problem,[166]

$$\begin{aligned}
J_{Ext-QUEST} = & \frac{1}{2} \bar{\mathbf{q}}^T \mathbf{K}^m \bar{\mathbf{q}} + \frac{1}{2} [\mathbf{R}_{qq} (\bar{\mathbf{q}} - \bar{\mathbf{q}}^-)]^T [\mathbf{R}_{qq} (\bar{\mathbf{q}} - \bar{\mathbf{q}}^-)] \quad (5.147) \\
& + \frac{1}{2} [\mathbf{R}_{\beta q} (\bar{\mathbf{q}} - \bar{\mathbf{q}}^-) + \mathbf{R}_{\beta\beta} (\boldsymbol{\beta} - \boldsymbol{\beta}^-)]^T \\
& \times [\mathbf{R}_{\beta q} (\bar{\mathbf{q}} - \bar{\mathbf{q}}^-) + \mathbf{R}_{\beta\beta} (\boldsymbol{\beta} - \boldsymbol{\beta}^-)]
\end{aligned}$$

where \mathbf{R}_{xx} are the square root information matrices that come from the left QR factorization of the information matrix at each epoch. The Extended-QUEST algorithm in Eq. 5.147 has been rewritten using the same nomenclature as in the SOAR Filter to facilitate easy comparison. Strong parallels between the SOAR Filter and Extended-QUEST are immediately evident by comparing the SOAR Filter objective function in Eq. 5.79 with the Extended-QUEST objective function in Eq. 5.147. These similarities produce similar results for the state vector update. The Extended-QUEST parameter vector update is given by

$$\hat{\boldsymbol{\beta}}^+ = \hat{\boldsymbol{\beta}}^- - \mathbf{R}_{\beta\beta}^{-1} \mathbf{R}_{\beta q} [\bar{\mathbf{q}} - \hat{\bar{\mathbf{q}}}^-] \quad (5.148)$$

which is clearly similar to the SOAR Filter parameter vector update from Eq. 5.104. The attitude update, however, is significantly different than in the SOAR Filter. Extended-QUEST finds the optimal *a-posteriori* attitude through the solution to

$$[\mathbf{K}^m + \mathbf{R}_{qq}^T \mathbf{R}_{qq}] \bar{\mathbf{q}} - \mathbf{R}_{qq}^T \mathbf{R}_{qq} \bar{\mathbf{q}}^- = \lambda \bar{\mathbf{q}} \quad (5.149)$$

The introduction of the constant vector $-\mathbf{R}_{qq}^T \mathbf{R}_{qq} \bar{\mathbf{q}}^-$ makes the solution to this problem more difficult than the simple eigenvector-eigenvalue problem in Eq. 5.103 for the SOAR Filter.

Despite strong similarities, there are numerous noteworthy differences between Extended-QUEST and the SOAR Filter. The first significant difference is that Extended-QUEST is a square-root information filter and, therefore, requires the use of the square-root information matrices that are obtained through QR factorization. Because the SOAR Filter uses the information matrix directly, QR factorization is not required. Although this reduces computations in the SOAR Filter, it places the SOAR Filter at a disadvantage from the standpoint of numerical precision relative to Extended-QUEST.

The second significant difference between the SOAR Filter and Extended-QUEST is how they express the attitude error. Extended-QUEST treats the attitude quaternion error from an additive standpoint, rather than a multiplicative standpoint. Although the term “additive” is used here, it is important to note that the Extended-QUEST algorithm explicitly enforces the quaternion unity norm constraint in generating the quaternion update. Therefore,

the update here should not be confused with additive updates that do not preserve the unity norm constraint.

The difference between using an additive or multiplicative quaternion update has two significant implications. The first is that the covariance associated with \mathbf{R}_{qq} is related to $\delta\bar{\mathbf{q}}_A = \bar{\mathbf{q}} - \hat{\mathbf{q}}^-$, rather than the multiplicative attitude quaternion error described in Eq. 5.40: $\delta\bar{\mathbf{q}} = \bar{\mathbf{q}} \otimes \hat{\mathbf{q}}^-$. This changes the physical significance of the resulting \mathbf{P}_{qq} in Extended-QUEST.

The second major implication of using the additive quaternion error is related to the optimal update. The additive representation of the attitude quaternion error requires the solution of a more general objective function. When the first differential of Eq. 5.147 is set to zero, Extended-QUEST requires the solution to Eq. 5.149, which is of the following form

$$(\mathbf{K} - \lambda\mathbf{I}_{4 \times 4})\bar{\mathbf{q}} + \mathbf{g} = \mathbf{0} \quad (5.150)$$

Psiaki demonstrates a robust method for finding the desired solution to this problem.[166] Solving Eq. 5.150, however, requires more computation than simply finding the largest eigenvalue/eigenvector of \mathbf{K} using a method such as QUEST or ESOQ. This means that solving for the optimal attitude in the SOAR Filter (as described in Eq. 5.103) will be faster than finding the optimal attitude in Extended-QUEST.

5.8 Example: Estimation of Attitude and Gyro Bias

Consider a scenario where a filter is required to estimate the attitude of a spacecraft and the gyro bias, \mathbf{b}_g . Rather than have the filter estimate the angular velocity, which would require models for external torques, the gyro measurements of angular velocity are taken to be truth. This is a common practice for spacecraft attitude filters - sometimes referred to as “flying the gyros.”

5.8.1 Theoretical Set-up for Example

Begin the derivation of this example SOAR error model by recalling the attitude error quaternion described in Eq. 5.40. Taking the time derivative of this equation,

$$\delta \dot{\mathbf{q}} = \dot{\mathbf{q}} \otimes \hat{\mathbf{q}}^{-1} + \bar{\mathbf{q}} \otimes \dot{\hat{\mathbf{q}}}^{-1} \quad (5.151)$$

Recalling the quaternion kinematic equation,

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \bar{\mathbf{q}} \quad \text{or} \quad \dot{\hat{\mathbf{q}}} = \frac{1}{2} \begin{bmatrix} \hat{\boldsymbol{\omega}} \\ 0 \end{bmatrix} \otimes \hat{\mathbf{q}} \quad (5.152)$$

and recognizing that $\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega}$, it is possible to show that to first order,[81]

$$\delta \dot{\mathbf{q}} = - \begin{bmatrix} [\hat{\boldsymbol{\omega}} \times] \delta \mathbf{q} \\ 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \delta \boldsymbol{\omega} \\ 0 \end{bmatrix} \quad (5.153)$$

Or equivalently,

$$\delta \dot{\mathbf{q}} = - [\hat{\boldsymbol{\omega}} \times] \delta \mathbf{q} + \frac{1}{2} \delta \boldsymbol{\omega} \quad (5.154)$$

$$\delta \dot{q}_4 = 0 \quad (5.155)$$

For small angles, $\delta\mathbf{q} \approx \delta\boldsymbol{\theta}/2$. Under these conditions Eq. 5.154 becomes

$$\delta\dot{\boldsymbol{\theta}} = -[\hat{\boldsymbol{\omega}} \times] \delta\boldsymbol{\theta} + \delta\boldsymbol{\omega} \quad (5.156)$$

All that is needed to obtain a final expression for the attitude error is a better understanding of $\delta\boldsymbol{\omega}$. Assume that the angular velocity is not a state to be estimated, but is a quantity provided by a rate-integration gyro. Further assume a gyro measurement model[178] given by

$$\tilde{\boldsymbol{\omega}} = [\mathbf{I}_{3 \times 3} + \mathbf{S}_g + \boldsymbol{\Gamma}_g] (\boldsymbol{\omega} + \mathbf{b}_g + \mathbf{w}_\omega) \quad (5.157)$$

where \mathbf{S}_g is the scale-factor error matrix, $\boldsymbol{\Gamma}_g$ is the misalignment error matrix, $\tilde{\boldsymbol{\omega}}$ is the measured angular velocity, \mathbf{b}_g is the gyro bias, and \mathbf{w}_ω is zero mean white noise. This general model is simplified by assuming that there is no scale-factor error, $\mathbf{S}_g = \mathbf{0}_{3 \times 3}$, and that there is no misalignment error, $\boldsymbol{\Gamma}_g = \mathbf{0}_{3 \times 3}$. Under these conditions, the gyro measurement model becomes

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{w}_\omega \quad (5.158)$$

The simplified measurement model of Eq. 5.158 is sometimes called Farrenkopf's gyro model and was first introduced in [179]. Further assume that the gyro bias behaves as a first-order Gauss-Markov process:

$$\dot{\mathbf{b}}_g = -\frac{1}{\tau} \mathbf{b}_g + \mathbf{w}_{\mathbf{b}} \quad (5.159)$$

where τ is the correlation time and $\mathbf{w}_{\mathbf{b}}$ is a zero mean white noise process. It may further be shown that the variance of $\mathbf{w}_{\mathbf{b}}$ is $2\sigma^2/\tau$. [152] This continuous-

time differential equation may be rewritten in discrete form as:[151, 152]

$$\mathbf{b}_g(t_{k+1}) = \exp \left[-\frac{t_{k+1} - t_k}{\tau} \right] \mathbf{b}_g(t_k) + \sigma \left\{ 1 - \exp \left[-2\frac{t_{k+1} - t_k}{\tau} \right] \right\}^{\frac{1}{2}} \mathbf{v}_1 \quad (5.160)$$

where $\mathbf{v}_1 \sim N(0, \mathbf{I}_{3 \times 3})$. Therefore,

$$\hat{\boldsymbol{\omega}} = E[\boldsymbol{\omega}] = \tilde{\boldsymbol{\omega}} - \hat{\mathbf{b}}_g \quad (5.161)$$

$$\hat{\dot{\mathbf{b}}}_g = E[\dot{\mathbf{b}}_g] = -\frac{1}{\tau} \hat{\mathbf{b}}_g \quad (5.162)$$

Noting that $\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \boldsymbol{\delta\omega}$,

$$\boldsymbol{\delta\omega} = -(\boldsymbol{\delta\mathbf{b}}_g + \mathbf{w}_\omega) \quad (5.163)$$

where $\boldsymbol{\delta\mathbf{b}}_g = \mathbf{b}_g - \hat{\mathbf{b}}_g$. Finally, substituting this into Eq. 5.156 yields the differential equation governing the attitude estimation error

$$\boldsymbol{\delta\dot{\boldsymbol{\theta}}} = -[\hat{\boldsymbol{\omega}} \times] \boldsymbol{\delta\boldsymbol{\theta}} - (\boldsymbol{\delta\mathbf{b}}_g + \mathbf{w}_\omega) \quad (5.164)$$

Similarly, the differential equation governing the bias estimation error is given by

$$\boldsymbol{\delta\dot{\mathbf{b}}}_g = -\frac{1}{\tau} \boldsymbol{\delta\mathbf{b}}_g + \mathbf{w}_b \quad (5.165)$$

Recalling the linear error model from Eq. 5.130,

$$\boldsymbol{\delta\mathbf{x}} = \begin{bmatrix} \boldsymbol{\delta\boldsymbol{\theta}}(t) \\ \boldsymbol{\delta\mathbf{b}}_g(t) \end{bmatrix} \quad (5.166)$$

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_\omega \\ \mathbf{w}_b \end{bmatrix} \quad (5.167)$$

$$\mathbf{F}(\hat{\mathbf{x}}, t) = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}(t) \times] & -\mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\frac{1}{\tau} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.168)$$

$$\mathbf{G}(t) = \begin{bmatrix} -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (5.169)$$

Additionally, let $\mathbf{Q}(t) = E[\mathbf{w}\mathbf{w}^T]$. If \mathbf{w}_ω and \mathbf{w}_b are assumed to be uncorrelated, it is straightforward to show that

$$\mathbf{Q}(t) = \begin{bmatrix} E[\mathbf{w}_\omega \mathbf{w}_\omega^T] & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & E[\mathbf{w}_b \mathbf{w}_b^T] \end{bmatrix} \quad (5.170)$$

5.8.2 Example Results

Two test cases are considered. The first is a scenario with typical sensor errors and noise. In the first scenario, only SOAR Filter errors and covariances are shown. The second scenario is a stressing case that contains very large *a-priori* attitude errors and bias errors. The measurement noise is also increased. Results for the second scenario are presented for the SOAR Filter, the MEKF, and Filter QUEST.

The first test case has performance specifications as described in Table 5.1. The results for this test case are shown in Fig. 5.2 and Fig. 5.3. Star tracker measurements were made available to the filter at a frequency of 1 Hz. The fact that almost all the results for the bias error lie within the $\pm 1 \sigma$ bounds indicates that the process noise could be reduced from the theoretically predicted value.

The second test case has performance specifications as described in Table 5.2. The results for this test case are shown in Fig. 5.4 and Fig. 5.5. The exact same noise and observations are applied to each filter technique

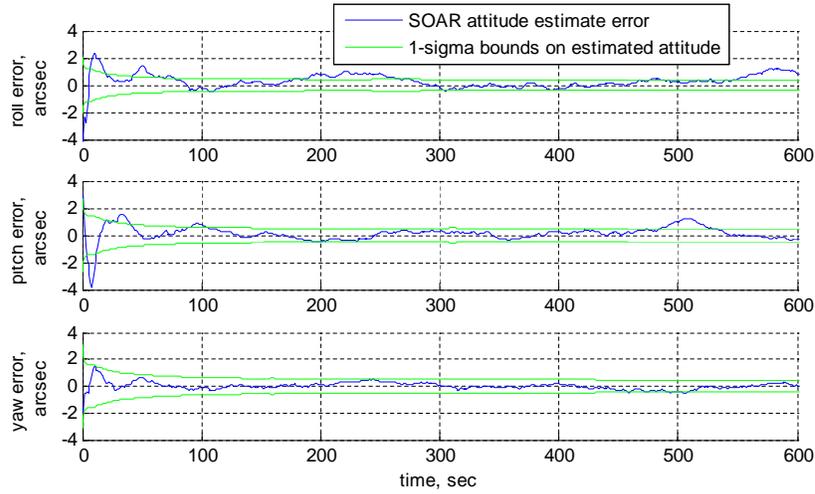


Figure 5.2: Estimated attitude errors for the SOAR Filter in scenario 1 (normal case with small errors).

in order to facilitate direct comparison. Under these extreme conditions, Filter QUEST is unable to converge because it is not estimating the gyro bias and because of its suboptimal fading memory factor. Both the MEKF and the SOAR Filter converge for this test case. Despite being given equivalent a-priori information and processing the exact same measurements, significant differences are observed prior to filter convergence. In all cases, the SOAR

Table 5.1: Summary of noise levels for SOAR Filter validations (scenario 1).

Description	Initial Conditions / Standard Deviation
<i>a-priori</i> attitude error	1 deg about $\mathbf{e}_\theta = [0.58 \ 0.58 \ 0.58]^T$
<i>a-priori</i> bias error	$\mathbf{b}_g - \hat{\mathbf{b}}_g = [0.25 \ -0.5 \ 0.375]^T$ deg/hr
Star observation error	$\sigma_{\phi,i} = 10$ arcsec
Gyro measurement error	$\sigma_\omega = 0.05$ deg/hr
Gyro bias Gauss-Markov error	$\sigma = 0.05$ deg/hr correlation time of $\tau = 1$ hr

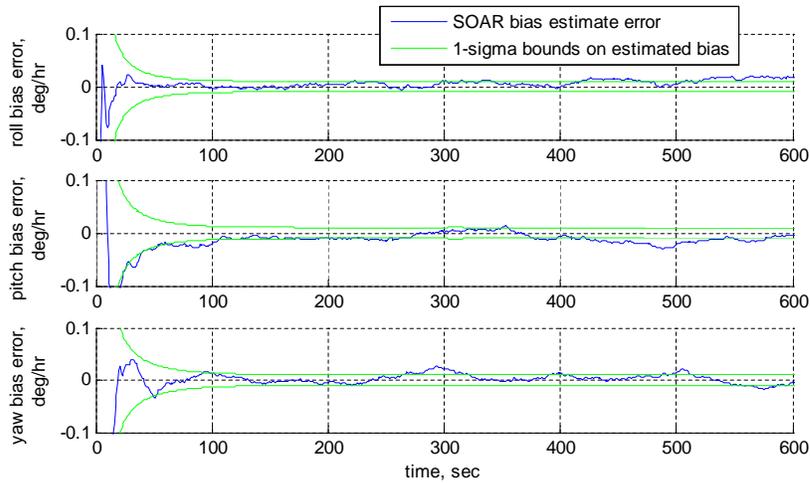


Figure 5.3: Estimated bias errors for the SOAR Filter in scenario 1 (normal case with small errors).

Filter performs as well or better than the MEKF. In both the attitude estimation and the bias estimation, the SOAR Filter has lower overshoot (the overshoot is induced by the very poor *a-priori* guess) and faster convergence. As was noted before, however, the SOAR Filter and the MEKF minimize the same objective function and are expected to exhibit similar behavior once the filter converges. The results shown in Fig. 5.4 support this observation. To make this observation clearer, a zoom-in of the roll-axis attitude error (see Fig. 5.6) shows that the SOAR Filter and the MEKF exhibit identical performance after the filter converges. Similar results may be seen for the other attitude errors and all of the bias errors.

Table 5.2: Summary of noise levels for SOAR Filter validations (scenario 2).

Description	Initial Conditions / Standard Deviation
<i>a-priori</i> attitude error	179 deg about $\mathbf{e}_\theta = [0.58 \ 0.58 \ 0.58]^T$
<i>a-priori</i> bias error	$\mathbf{b}_g - \hat{\mathbf{b}}_g = [90 \ -90 \ 67.5]^T$ deg/hr
Star observation error	$\sigma_{\phi,i} = 1$ deg
Gyro measurement error	$\sigma_\omega = 0.5$ deg/hr
Gyro bias Gauss-Markov error	$\sigma = 0.5$ deg/hr correlation time of $\tau = 1$ hr

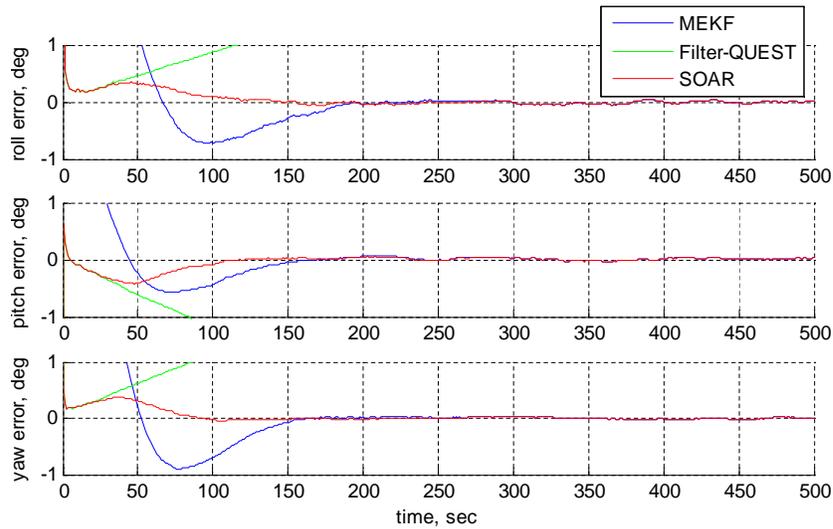


Figure 5.4: Comparison of the estimated attitude errors for the MEKF, Filter QUEST, and the SOAR Filter in in scenario 2 (stressing case with large errors).

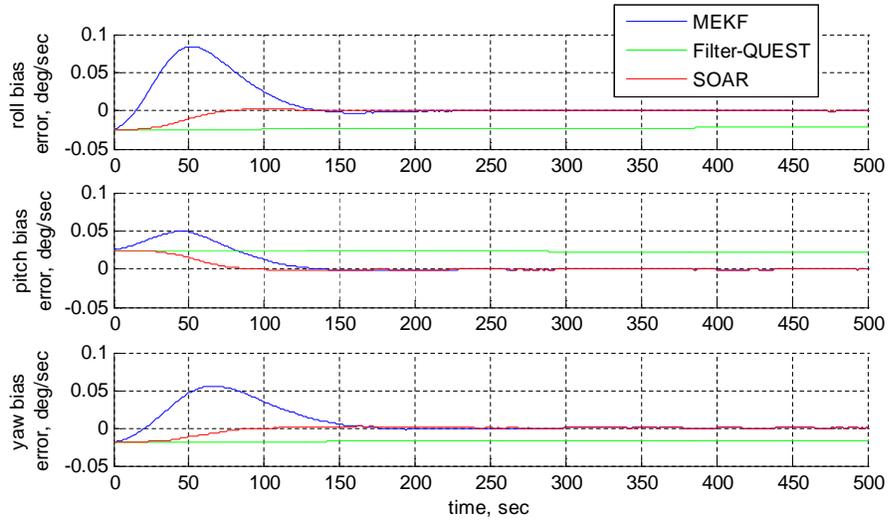


Figure 5.5: Comparison of the estimated bias errors for the MEKF, Filter QUEST, and the SOAR Filter in in scenario 2 (stressing case with large errors).

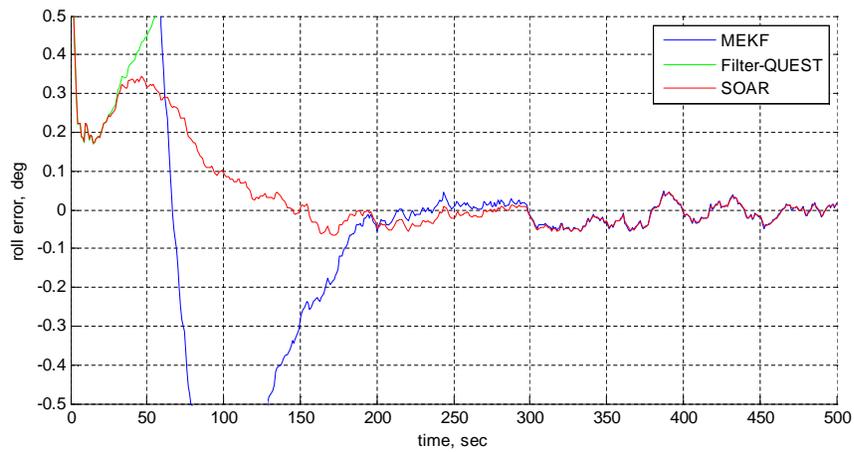


Figure 5.6: Zoom-in of roll-axis attitude error for scenario 2 (stressing case with large errors).

Chapter 6

Spacecraft Optical Navigation

This chapter provides two case study examples of the estimation of position, velocity, and attitude for a spacecraft using the optical navigation measurements given in Chapter 4. Spacecraft position and velocity are estimated using an Extended Kalman Filter (EKF) and the attitude is estimated using the newly developed SOAR Filter, which was introduced in Chapter 5.

This chapter begins with a very brief review of the EKF. Next, in Section 6.2, the dynamics model used to propagate the spacecraft trajectory is presented. Because this model requires ephemeris information for each of the simulated planets, the JPL ephemeris files are introduced in Section 6.3. With these preliminary concepts complete, the last two sections focus on the two case studies.

The first case study, presented in Section 6.4, demonstrates optical navigation for a robotic spacecraft performing a fly-by of Venus. This case study uses real images collected by the MESSENGER spacecraft during its June 2007 fly-by of this planet.

The second case study, presented in Section 6.5, demonstrates optical navigation for a crewed vehicle performing an autonomous lunar return

in February 2024. In this case, it is assumed that the spacecraft has lost contact with Earth and must meet the vehicle reentry constraints without using ground-based navigation resources. Synthetic images of both the Earth and Moon are generated and used to assess the performance of a completely autonomous lunar return with only optical navigation measurements.

6.1 Review of the Extended Kalman Filter (EKF)

The Kalman Filter, originally developed in the early 1960s,[180, 181] is now one of the most common techniques used for real-time navigation. The Kalman Filter is a recursive estimation algorithm that provides a minimum variance estimate of some state, \mathbf{x} , given a set of measurements, \mathbf{y} . The Kalman Filter assumes linear dynamics such that

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w} \quad (6.1)$$

where \mathbf{w} is zero mean white noise. It also assumes a linear measurement model,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (6.2)$$

where \mathbf{v} is zero mean white noise. Unfortunately, the present problem has both nonlinear dynamics and nonlinear measurements. One of the most common methods for dealing with this type of problem is through the EKF, which linearizes the state error and measurement residual about the best estimate of the state. The derivation of the EKF is well known and is omitted here for brevity. Detailed discussions of the EKF may be found in numerous different references, including [150], [151], [152], and [182]

The version of the EKF used here is the continuous-discrete EKF. This means that the measurement updates occur at discrete times, but the trajectory and covariance is propagated in a continuous fashion between these updates. What follows is a brief summary of the key equations in the continuous-discrete EKF.

Suppose the dynamics of a system are governed by the following nonlinear model,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{G}(t)\mathbf{w}(t) \quad (6.3)$$

Further assume that the measurements are given by some nonlinear measurement model,

$$\tilde{\mathbf{y}} = \mathbf{h}(\mathbf{x}(t), t) + \mathbf{v}(t) \quad (6.4)$$

Linearizing about the reference trajectory, define the the Jacobian matrices

$$\mathbf{F} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}} \quad \mathbf{H} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}} \quad (6.5)$$

where \mathbf{H} is the measurement sensitivity matrix discussed in the last section of Chapter 4. Therefore, in the EKF, the state update given a new measurement, $\tilde{\mathbf{y}}_i$, taken at time t_i is given by

$$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \mathbf{K}_i [\tilde{\mathbf{y}}_i - \mathbf{h}(\hat{\mathbf{x}}_i^-, t_i)] \quad (6.6)$$

and the covariance update is given by

$$\mathbf{P}_i^+ = [\mathbf{I}_{n \times n} - \mathbf{K}_i \mathbf{H}_i] \mathbf{P}_i^- \quad (6.7)$$

where \mathbf{K}_i is the Kalman gain and is given by

$$\mathbf{K}_i = \mathbf{P}_i^- \mathbf{H}_i^T [\mathbf{H}_i \mathbf{P}_i^- \mathbf{H}_i^T + \mathbf{R}_i]^{-1} \quad (6.8)$$

Note, however, that it is sometimes advantageous to use the Joseph form of the covariance update

$$\mathbf{P}_i^+ = [\mathbf{I}_{n \times n} - \mathbf{K}_i \mathbf{H}_i] \mathbf{P}_i^- [\mathbf{I}_{n \times n} - \mathbf{K}_i \mathbf{H}_i]^T + \mathbf{K}_i \mathbf{R}_i \mathbf{K}_i^T \quad (6.9)$$

which is valid for any gain matrix \mathbf{K}_i . Although Eq. 6.7 and Eq. 6.9 are mathematically identical if the optimal \mathbf{K}_i is chosen, the Joseph form of the covariance update has the advantage of introducing less error into \mathbf{P}_i^+ if there are errors in \mathbf{K}_i . The Joseph form, however, is more computationally expensive than the update given in Eq. 6.7.

After the update, the state is propagated to time t_{i+1} using Eq. 6.3. The covariance is also propagated to the next measurement time using

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}, t) \mathbf{P}(t) + \mathbf{P}(t) \mathbf{F}(\hat{\mathbf{x}}, t)^T + \mathbf{G}(t) \mathbf{Q}(t) \mathbf{G}(t)^T \quad (6.10)$$

6.2 Translational Spacecraft Dynamics Model

The translational dynamics model used here assumes that gravity is the only force acting on the spacecraft. Therefore, the effects of various non-conservative environmental forces (e.g. solar radiation pressure, waste venting, drag, etc.) are not considered. Additionally, it is assumed that the Sun and planets act as point masses. This reduces the spacecraft translational dynamic problem to the classic n -body orbital mechanics problem. This problem is discussed thoroughly in the literature. Especially nice presentations of this material may be found in [153] and [183].

The classic result for the acceleration experienced by a spacecraft in an n -body system, which is presented without derivation for brevity, is given by

$$\ddot{\mathbf{r}}_{sc} = -\mu_j \frac{\mathbf{r}_{sc}}{\|\mathbf{r}_{sc}\|^3} + \sum_{\substack{k=1 \\ k \neq j}}^n \mu_k \left(\frac{\mathbf{r}_k - \mathbf{r}_{sc}}{\|\mathbf{r}_k - \mathbf{r}_{sc}\|^3} - \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|^3} \right) \quad (6.11)$$

where body j is the central body, \mathbf{r}_{sc} is the position of the spacecraft with respect to the central body, \mathbf{r}_k is the position of the k -th celestial body with respect to the central body, and μ_k is the gravitational constant of the k -th celestial body. It is important to note that the trajectories of the $n - 1$ perturbing bodies are not propagated in this simulation. Rather, the position of any celestial body at any point in time is obtained from the JPL ephemeris files discussed in Section 6.3.

To use the EKF, it is also necessary to compute the Jacobian of the translational state vector. If the state vector for the translational states is given by

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{sc} \\ \dot{\mathbf{r}}_{sc} \end{bmatrix} \quad (6.12)$$

then the Jacobian from Eq. 6.5, $\mathbf{F} = \partial \dot{\mathbf{x}} / \partial \mathbf{x}$, is well known to be

$$\mathbf{F} = \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{U} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (6.13)$$

where

$$\begin{aligned} \mathbf{U} = & -\frac{\mu_j}{\|\mathbf{r}_{sc}\|^3} \mathbf{I}_{3 \times 3} + \frac{3\mu_j}{\|\mathbf{r}_{sc}\|^5} \mathbf{r}_{sc} \mathbf{r}_{sc}^T \\ & - \sum_{\substack{k=1 \\ k \neq j}}^n \mu_k \left(\frac{1}{\|\mathbf{r}_k - \mathbf{r}_{sc}\|^3} \mathbf{I}_{3 \times 3} + \frac{3}{\|\mathbf{r}_k - \mathbf{r}_{sc}\|^5} (\mathbf{r}_k - \mathbf{r}_{sc}) (\mathbf{r}_k - \mathbf{r}_{sc})^T \right) \end{aligned} \quad (6.14)$$

6.3 Jet Propulsion Laboratory Ephemeris Files

JPL maintains a set of high-precision ephemerides for a number of bodies in our solar system. Various types of ephemeris files are available, but the DE405 ephemeris files are used here. More information about the JPL DE405 ephemeris files may be found in [184].

The DE405 ephemeris files may be easily obtained through the JPL Solar System Dynamics website. The ephemeris files typically obtained from JPL are ASCII files and are usually available in 20-year blocks. It is frequently convenient to combine multiple blocks (if ephemerides are required over a time interval that spans more than one of the 20-year blocks) into a contiguous block. The ASCII files must also be converted into a binary format for use with the ephemeris generation software. The ephemeris generating software used here is the MATLAB Based Solar System Ephemeris Toolbox developed at the NASA Kennedy Space Center.¹

6.4 Case Study 1: Optical Navigation of a Robotic Spacecraft during a Venus Fly-by

Consider a robotic spacecraft performing a planetary fly-by as the first of two different case studies. This case study is used to highlight the major issues associated with optical navigation during a planetary encounter and provide a representative sample of the navigation performance achievable using

¹The MATLAB Based Solar System Ephemeris Toolbox is publicly available through the Open Channel Foundation at <http://www.openchannelsoftware.com/>

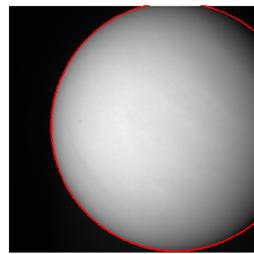
only optical navigation. The MESSENGER spacecraft's June 2007 fly-by of Venus is used to help provide concrete mission context. During this fly-by, numerous images were taken by the wide angle camera on the Mercury Dual Imaging System (MDIS). In this case study, the actual images acquired by the MDIS are used.² Although both raw and calibrated images are available, only the raw images are used in the subsequent analyses. It is important to note that all images are stored as expressed in the frontal image plane (see Fig. 3.7). Therefore, as is discussed at the end of Section 3.3, the raw image file must be flipped vertically (i.e. first row becomes the last row, and last row becomes the first row) to transform the image from the frontal image plane to the actual detector plane. Additionally, each image is accompanied by a label file that contains useful data such as: PDS product ID; date and time that the image was taken; camera exposure duration; estimated spacecraft position, velocity, and attitude; camera resolution; and much more. A complete discussion of the label file for the Engineering Data Record (EDR) images may be found in [185]. For the remainder of this discussion, the position, velocity, and attitude estimates obtained from these label files are assumed to represent truth. Therefore, the state errors presented below actually represent deviations from the state estimated by the MESSENGER navigation team.

Many images of Venus were taken during the 5-6 June 2007 fly-by. Images were acquired by both the wide angle camera and the narrow angle

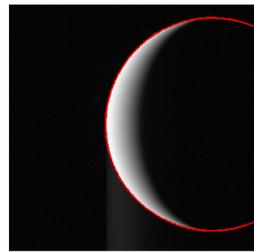
²These images are publicly available and may be obtained through the NASA Planetary Data System's Geosciences Node operated by Washington University in St. Louis. The PDS Geosciences Node may be found online at <http://ode.rs1.wustl.edu/>.

camera. Here, only the images taken by the wide angle camera will be used for optical navigation. Further, as the MESSENGER spacecraft approached the periapsis of the Venus fly-by trajectory, the planet became too large to fit entirely within the camera FOV. Images acquired during this period are omitted from the optical navigation procedure because a sufficient portion of the planet horizon is not visible in the image. After these constraints are applied, a total of 13 images remain that are suitable for optical navigation. The location of the spacecraft on the fly-by trajectory when of all 13 of these images were taken is shown in Fig. 6.1. The first image was taken before the fly-by and the remaining 12 images were taken after the fly-by. The last 12 images are equally spaced in time with 1 hour between each image. Note that Fig. 6.1 also contains three example images with the best fit ellipse from the image processing algorithm developed in Chapter 4 superimposed in red. A brief summary of some relevant details for all 13 images is provided in Table 6.1.

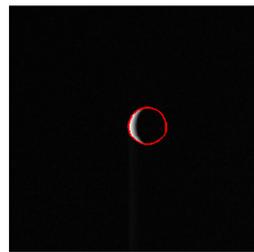
These 13 images are used to generate the optical navigation measurements discussed in Chapter 4. Although Fig. 6.1 lists the measurement error for a few different cases, the measurement errors for all 13 cases are shown graphically in Fig. 6.2. The top plot in this figure shows the absolute error (with respect to the estimated state from the MESSENGER image label file) and the bottom plot normalizes this error with respect to range between the spacecraft and Venus. First, note that the magnitude of the measurement error in the radial direction is significantly larger than in the crosstrack di-



a) Image ID: 1
 Date: 05 June 2007 Time: 21:18:29.737 UTC
 PDS Product ID: ew0089565636k
 Image Size: 1024 x 1024 pixels
 Measurement Error:* 88.1 km (radial direction)
 44.8 km (transverse direction)



b) Image ID: 2
 Date: 06 June 2007 Time: 01:18:19.750 UTC
 PDS Product ID: ew0089580026f
 Image Size: 512 x 512 pixels (binned to reduce image size)
 Measurement Error:* -370.8 km (radial direction)
 55.1 km (transverse direction)



c) Image ID: 13
 Date: 06 June 2007 Time: 12:18:19.746 UTC
 PDS Product ID: ew0089619626f
 Image Size: 512 x 512 pixels (binned to reduce image size)
 Measurement Error:* -4029.1 km (radial direction)
 169.7 km (transverse direction)

*Measurement error is taken as the difference between the measured position generated from the image processing algorithm and the position estimate from the MESSENGER image label file.

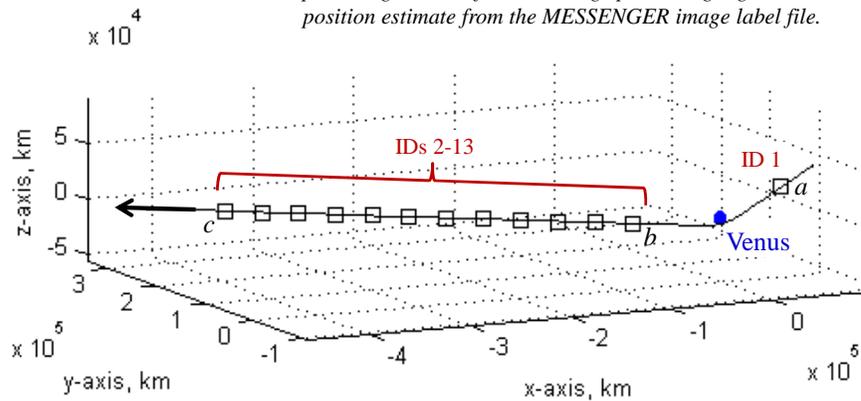


Figure 6.1: Image geometry for June 2007 MESSENGER fly-by of Venus. The black boxes on the trajectory diagram denote the spacecraft location at the time that each of the 13 images were acquired. The example raw images a-c are superimposed with red ellipses indicating the horizon location estimated by the image processing algorithm. Raw images are from [93].

Table 6.1: Summary of MESSENGER images used in Venus fly-by case study. All images were taken with the MDIS wide angle camera.

Image ID	PDS Product ID	Date	Time (UTC)	Image Resolution
1	ew0089565636k	5 June 2007	21:18:29.737	1024 × 1024
2	ew0089580026f	6 June 2007	01:18:19.750	512 × 512
3	ew0089583626f	6 June 2007	02:18:19.759	512 × 512
4	ew0089587226f	6 June 2007	03:18:19.758	512 × 512
5	ew0089590826f	6 June 2007	04:18:19.747	512 × 512
6	ew0089594426f	6 June 2007	05:18:19.747	512 × 512
7	ew0089598026f	6 June 2007	06:18:19.747	512 × 512
8	ew0089601626f	6 June 2007	07:18:19.747	512 × 512
9	ew0089605226f	6 June 2007	08:18:19.758	512 × 512
10	ew0089608826f	6 June 2007	09:18:19.757	512 × 512
11	ew0089612426f	6 June 2007	10:18:19.746	512 × 512
12	ew0089616026f	6 June 2007	11:18:19.747	512 × 512
13	ew0089619626f	6 June 2007	12:18:19.746	512 × 512

rection. This is as expected given the pencil-shaped covariance discussed in Section 4.9 (and shown pictorially in Fig. 4.27).

The speed at which these measurements may be made depends on the resolution of the image (in this case either a 1024×1024 or a 512×512 image) and the size of the body in the image. The image processing algorithm will run faster if the observed object is smaller (takes up fewer pixels). The run times seen for processing each of the 13 images in this case study are shown in Table 6.2. The time required to generate the initial measurement and to generate the refined measurement are both provided. Note that these run times are for the image processing algorithm as implemented in MATLAB (not compiled to C) and no attempt has been made to optimize the algorithm

Table 6.2: Summary of image processing run times required to generate optical navigation measurements. These run times correspond to the image processing algorithm implemented in MATLAB V 7.8.0.347 (R2009a) and run on a laptop computer with Windows Vista, a 2.0 GHz Intel Core 2 Duo CPU, and 4 GB of RAM.

Image ID	Image Processing Run Time	
	Initial Measurement	Refined Measurement
1	72.744 sec	209.176 sec
2	11.101 sec	29.811 sec
3	8.399 sec	21.475 sec
4	6.214 sec	15.155 sec
5	5.473 sec	12.173 sec
6	4.577 sec	10.367 sec
7	4.173 sec	9.161 sec
8	4.171 sec	8.436 sec
9	3.939 sec	7.771 sec
10	3.316 sec	7.411 sec
11	3.128 sec	6.959 sec
12	2.979 sec	6.696 sec
13	2.808 sec	6.428 sec

or code implementation for speed.

Now, consider a scenario where the initial conditions of the spacecraft at the first image location (see Table 6.3) have a 1σ position error of 1 km, and a 1σ velocity error of 2 m/s. These states may be propagated forward in time using the EOM discussed in Section 6.2. In this simulation, only gravitational effects from Venus and the Sun are considered, with planetary ephermedies obtained from the JPL DE405 ephemeris files. If no process noise is assumed, the results are as shown in Fig. 6.3. This figure shows the covariance of the final state with no measurements and the covariance of the final state with

an EKF that only uses the 13 optical navigation measurements. Results of a Monte Carlo analysis are superimposed on these covariance ellipsoids and demonstrate good agreement.

The displacement of these covariance ellipses (and of the Monte Carlo results) from the spacecraft location estimated by the MESSENGER navigation team (blue dot at origin of Fig. 6.3) indicates that the simple n -body problem from Section 6.2 does not capture all the dynamics. This, of course, is as expected. Therefore, process noise is added to account for these unmodeled dynamics. Here, it was found that simply adding diagonal process noise

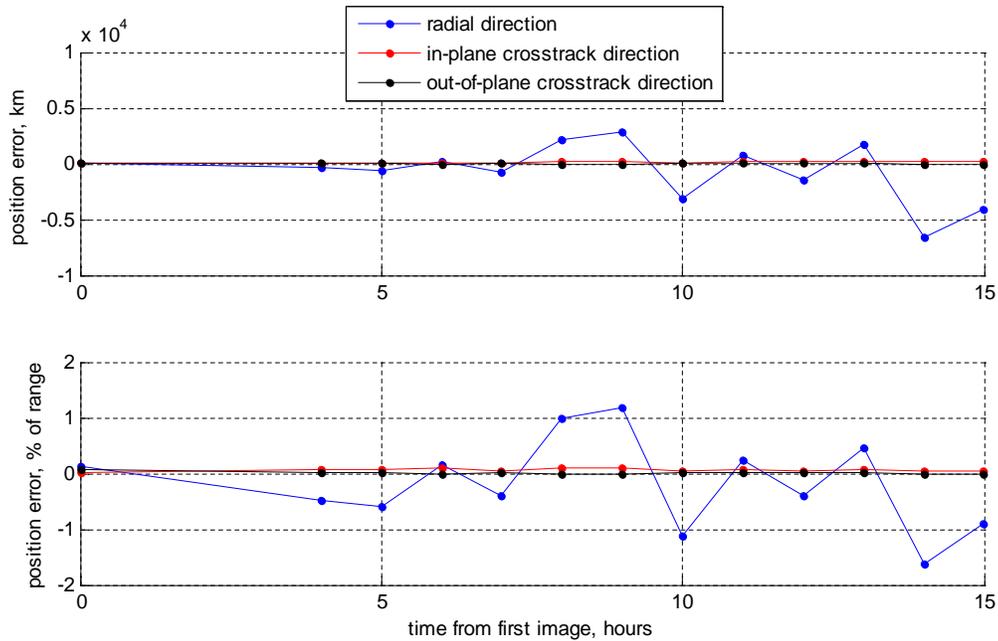


Figure 6.2: Measurement error from instantaneous position fix generated from apparent diameter and centroid of Venus.

Table 6.3: Initial conditions for MESSENGER fly-by of Venus at 21:18:29.737 UTC on 5 June 2007. These quantities are expressed in a Venus-relative reference frame.

Quantity	x-direction	y-direction	z-direction
Position	62372.8 km	3343.4 km	23175.5 km
Velocity	-8.536 km/s	-1.557 km/s	-3.934 km/s

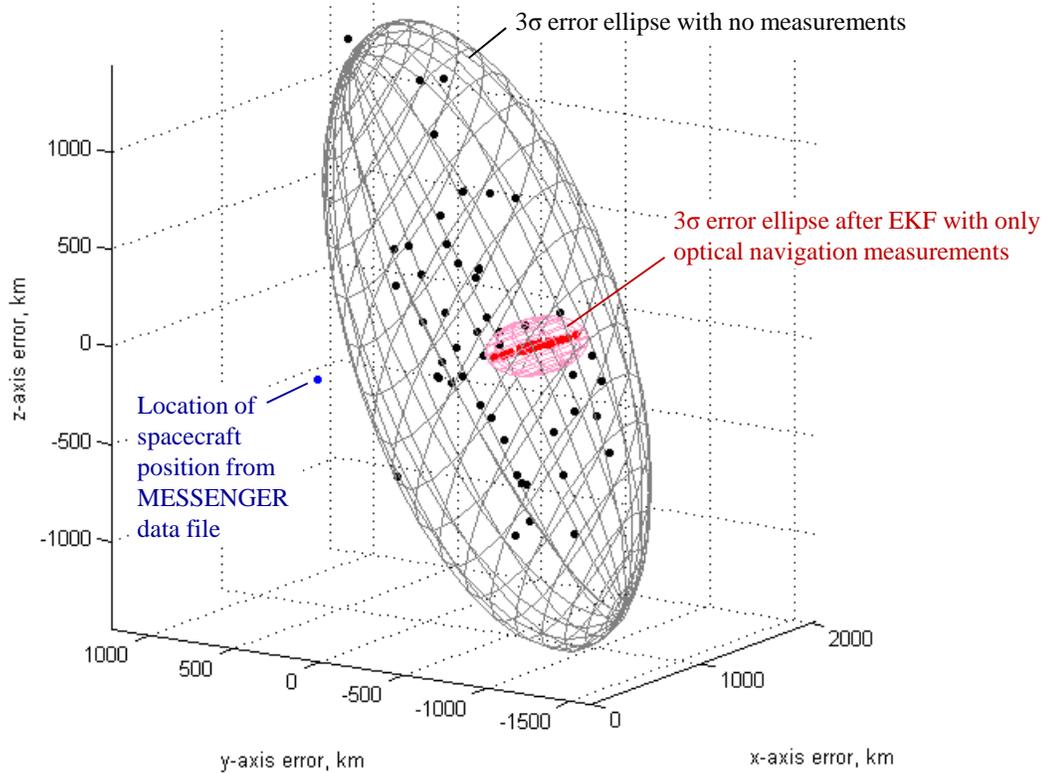


Figure 6.3: Spacecraft position covariance at the end of the Venus fly-by (location of last image). Covariance is shown for a fly-by with no measurements (gray ellipse and black dots for Monte Carlo results) and a fly-by that uses only optical navigation measurements in an EKF (pink ellipse and red dots for Monte Carlo results). Results assume no process noise.

to the acceleration terms provided good results,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_a^2 \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (6.15)$$

where σ_a is the standard deviation of the process noise on the acceleration term.

Once process noise is added, the position and velocity errors are as shown in Fig. 6.4 and Fig. 6.5, respectively. The addition of process noise allows for much of the offset seen in the end-state position estimates shown in Fig. 6.3 to be removed. Further, a quick comparison between Fig. 6.2 with Fig. 6.4 shows that the filtered position error is an order of magnitude smaller than the measurement errors. The filtered position and velocity errors fall within the 1σ covariance bounds for most of the trajectory, and fall within the 3σ covariance bounds for all points.

It is interesting to note the continued growth of the covariance in the radial direction (for both position and velocity). The covariance bounds in the radial direction do not show the same level of reduction with new measurements as is seen in the crosstrack directions. This result is easy to explain. First, growth of the covariance between measurement times is expected from the covariance propagation equation shown in Eq. 6.10. Unfortunately, as is evident from Eq. 4.107, the measurement covariance in the radial direction also grows with distance from the target. This means that each successive optical navigation measurement from Image ID 2-13 has a larger and larger measurement covariance in the radial direction. These effects combine to cre-

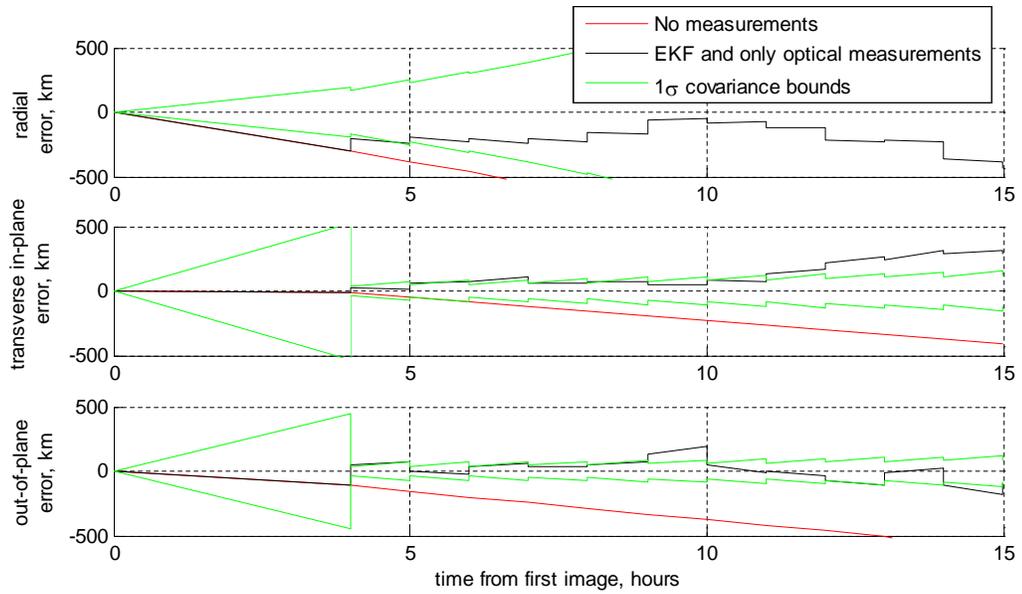


Figure 6.4: Position error for June 2007 MESSENGER Venus fly-by.

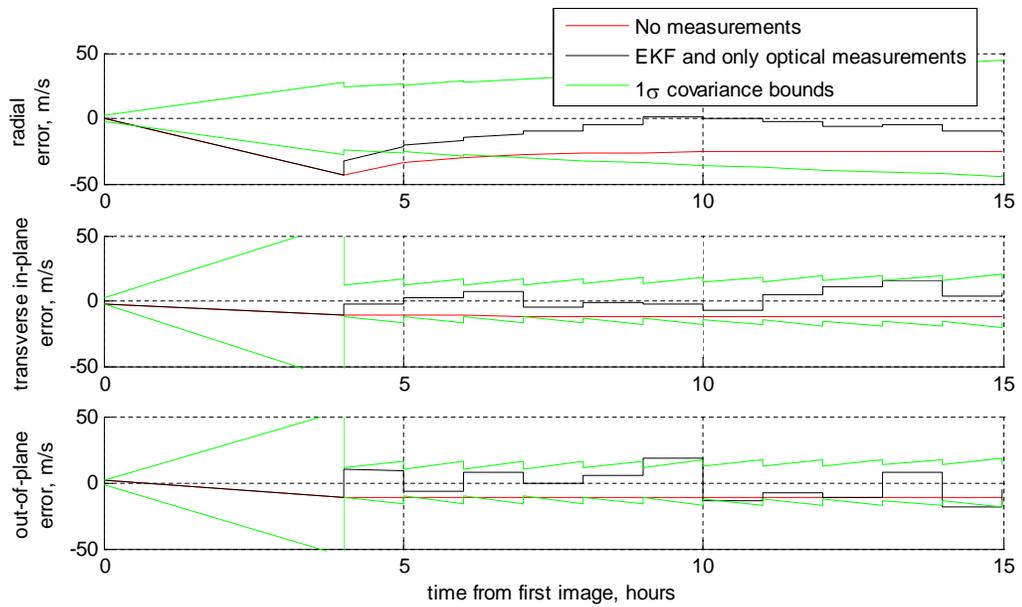


Figure 6.5: Velocity error for June 2007 MESSENGER Venus fly-by.

ate a continued growth of the radial covariance as the spacecraft flies farther away from Venus.

The performance of the optical navigation system in this situation is limited by a number of different factors. First, the geometry of this fly-by trajectory after the spacecraft passes periapsis is poor - a problem that is likely to be encountered in most fly-by scenarios. As is shown from the trajectory diagram at the bottom of Fig. 6.3, the spacecraft moves in nearly a straight line away from the planet, meaning that the spacecraft is moving primarily along radial direction. This means that the pencil-shaped measurement covariance will have (approximately) the same orientation from image to image. The end result is a poorer estimate of position and velocity in the radial direction than would be possible in a scenario that exhibits significant changes in image geometry.

The second major issues limiting performance is image resolution. With the exception of the first image, all of the images used in this analysis had a resolution of 512×512 pixels. Although the detector on MESSENGER's MDIS wide angle camera was a 1024×1024 CCD, the image was frequently binned into a 512×512 array by combining four adjacent pixels into one pixel. This binning was required to reduce the image size so that the 1 Hz data throughput requirement could be satisfied (full frame images could only be sent at 0.25 Hz).[125] This reduced resolution was acceptable because spatial sampling requirements could be met with only a 512×512 image.[125] Although a 512×512 image may meet science spatial sampling requirements, the reduced

resolution degrades the accuracy of the optical navigation measurements that may be extracted from the image. In the wide angle camera, for example, an error of 1/10 pixel corresponds to a line-of-sight direction error of about 3.7 arcsec for the 1024×1024 image, but increases to 7.4 arcsec for the binned 512×512 image.

A third source of error is the atmosphere of Venus. Optical navigation measurements that rely on accurate identification of the horizon of a target with an atmosphere is a difficult task. This difficulty has been experienced by both humans and robotic missions. Recall from earlier discussions that astronauts taking measurements with a space sextant could only locate a point in the blurry Earth horizon with a repeatability of a few kilometers.[28] Alternately, and of particular relevance here, the MESSENGER optical navigation team did not use optical navigation during the Earth or Venus fly-bys primarily due to difficulties of accurately determining the horizon in the presence of an atmosphere.[66] In the present work, it was found that good results could be achieved if the radius of Venus was increased by about 75-115 km, which is commensurate with the altitude of the upper cloud layer on Venus.

6.5 Case Study 2: Human Cislunar Navigation

As a second motivating example, consider a crewed spacecraft on a lunar return trajectory. With crew onboard the vehicle, it is likely that the ability to return to Earth independent of ground-based tracking will be required.[186] It is worth noting here that the problem of an autonomous lunar

return has been an active area of research in recent years. Some of the notable recent works include [35] and [187], which provide additional details about the problem and discuss alternative approaches.

For this example scenario, suppose that the crewed spacecraft performs a three-burn Trans-Earth Injection (TEI) sequence. The return trajectory occurs from 19-24 February 2024 and the spacecraft states after each of the three burns are as shown in Table 6.4.³ Further suppose that communication between Earth and the spacecraft is lost after the third burn. This means that the vehicle must perform autonomous navigation from this point until entry interface (typically about 3.5-4 days for most return trajectories). In the present study the autonomous return is performed using only optical navigation measurements. Because no real images of the Earth and Moon throughout an entire lunar return were available, synthetic images of the Earth and Moon were created using the techniques outlined in Section 3.9.2.

For a situation of this type, crew safety is of primary importance and other requirements associated with nominal operations may be relaxed. The most important parameter for Earth re-entry is the entry flight path angle (FPA). The reference trajectory described in Table 6.4 has an entry velocity of 10.985 km/s with an entry FPA of -5.86° . For Apollo, the total entry FPA error was required to be less than 1.0° - and only half of this error (0.5°) could be from navigation related errors.[187] For more complicated re-entry

³This reference trajectory was generously provided by Sara Scarritt at the Department of Aerospace Engineering and Engineering Mechanics at The University of Texas at Austin.

Table 6.4: Summary of spacecraft states on lunar return trajectory after each of the three TEI maneuvers.

Quantity		TEI 1	TEI 2	TEI 3
Date (in year 2024)		February 19	February 20	February 20
Time (UTC)		03:27:10.4	01:23:22.6	11:04:15.5
Position	x-axis	2290.1 km	-78060.1 km	-109299.0 km
	y-axis	346576.6 km	331573.6 km	337307.4 km
	z-axis	184942.7 km	209991.8 km	179256.2 km
Velocity	x-axis	1.2305 km/s	-1.1858 km/s	0.5123 km/s
	y-axis	0.1275 km/s	0.1040 km/s	0.0686 km/s
	z-axis	0.1815 km/s	-0.6233 km/s	-0.8046 km/s

scenarios, the entry FPA requirement is expected to be much tighter (e.g. an FPA error less than 0.1° is required for the skip entry planned for Orion [35]). Even for a vehicle such as the Orion, which nominally plans to have a skip entry, it may be possible to relax this requirement in the event of a communication system failure by performing a different type of entry. These off-nominal entries would likely come at the cost of decreased performance (e.g. less control over landing ellipse, higher deceleration experienced by crew, etc.). Therefore, the remainder of this case study assumes that the entry FPA navigation error must be less than 0.5° .

6.5.1 Attitude Estimation

For this scenario, assume that the vehicle is equipped with two star trackers with a $20^\circ \times 20^\circ$ FOV. Each star tracker is further assumed to have a 1024×1024 pixel CCD detector. The star trackers are mounted perpendicular to one another. One star tracker has the camera boresight pointing along the

vehicle y-axis, and the other has the camera boresight pointing along the vehicle z-axis. Each of these star trackers use the QUEST algorithm to compute an attitude quaternion using stars within the camera FOV. The quaternions generated by these star trackers are supplied to the attitude filter at a rate of 1 Hz.

It is also assumed that the vehicle is equipped with an IMU similar to Honeywell's MIMU (see [188] for detailed instrument specifications). This device provides data to the attitude filter at 40 Hz. As was done in the examples of Chapter 5, the angular velocity measurements from the gyro are taken as truth (angular velocity is not estimated). The gyro bias is modeled as a first-order Gauss-Markov process and is estimated in real-time by the attitude filter onboard the spacecraft.

These measurements are processed by the newly developed SOAR Filter. For a rigid-body spacecraft rotating with an initial angular velocity of 1 deg/sec, the results of first hour of attitude filtering are shown in Fig. 6.6 and Fig. 6.7. These results clearly indicate that a steady-state attitude covariance is quickly reached.

6.5.2 Position and Velocity Estimation

The lunar return case study discussed here provides a different perspective than what was seen in the Venus case study. Many of the difficulties experienced in the Venus fly-by are no longer present. These differences result in a noticeable improvement for the lunar return example, despite a similar

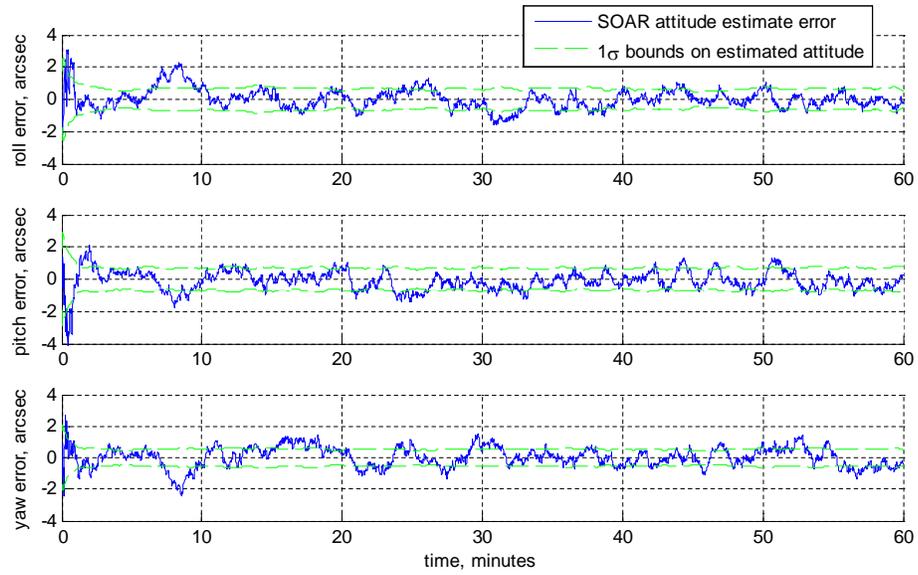


Figure 6.6: Attitude error from SOAR Filter.

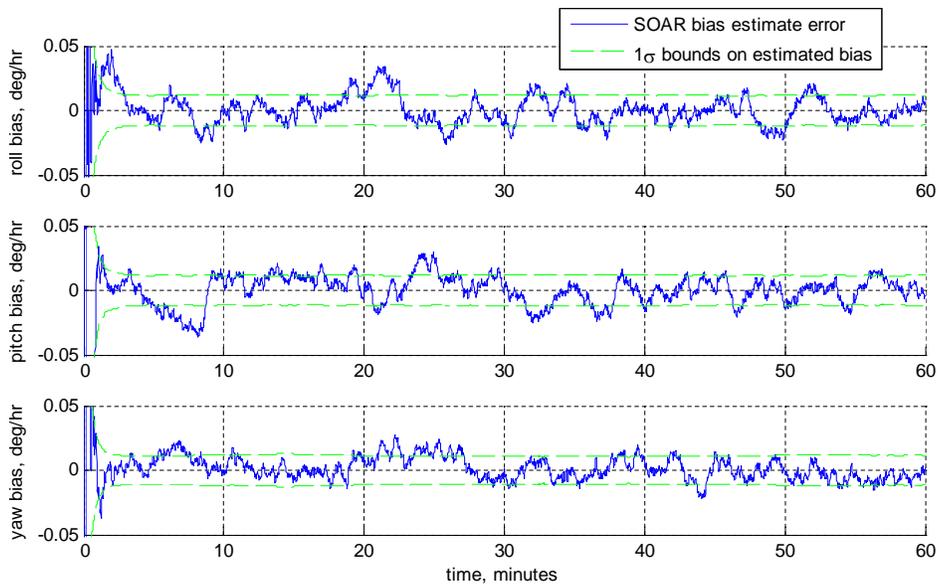


Figure 6.7: Gyro bias error from SOAR Filter.

level of measurement noise. The most important difference is the viewing geometry throughout the trajectory. Recall that the Venus optical navigation images were primarily taken after the fly-by as the spacecraft moved in nearly a straight line away from the planet. For the case of the lunar return trajectory, shown in Fig. 6.8, there is a changing geometry between the spacecraft trajectory and Earth. Additionally, the spacecraft may image the Moon during the return, supplying an optical observation that can become nearly perpendicular in direction to the line-of-sight to the Earth.

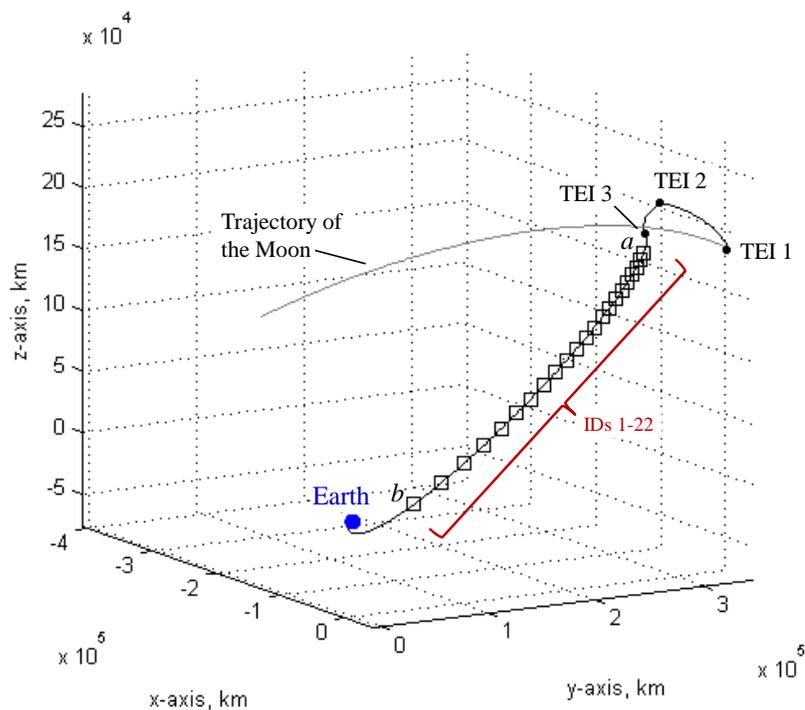
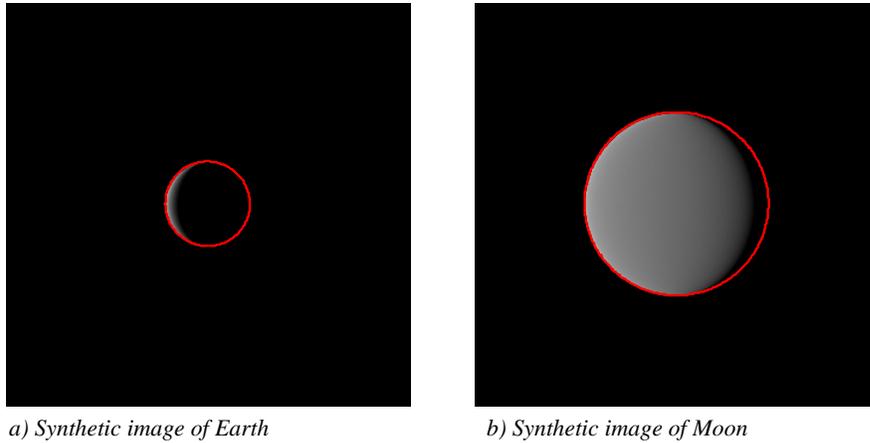


Figure 6.8: Image geometry for 19-24 February 2024 lunar return trajectory with a three-burn TEI sequence. The black boxes along the spacecraft return trajectory denote the 22 different locations at which images are acquired.

During the lunar return, images of the Earth and Moon were taken at 22 different locations (resulting in a total of 44 images). These 22 imaging locations are spaced equally in time (four hours between each imaging location) and all occur on the final return arc after the third TEI burn. The camera used to capture these images is assumed to have a $9.2^\circ \times 9.2^\circ$ FOV and a 1024×1024 CMOS detector. As an example, the synthetic images from the first image location (ID 1) and the last image location (ID 22) are provided in Fig. 6.9 and Fig. 6.10, respectively. The instantaneous position fix measurement error (from apparent diameter and centroid) for all 22 of the Earth images is shown in Fig. 6.11. These results are also shown for all 22 of the Moon images in Fig. 6.12.

It is interesting to compare the results of Fig. 6.11 with those seen in the Venus fly-by (see Fig. 6.2). This is a reasonable comparison because Earth and Venus have a similar diameter, and both have an atmosphere. A quick inspection will show that the images of Earth, on average, have a slightly lower error. This is to be expected because the images of Earth are at higher resolution. The minimum angular width of a single pixel in the MDIS wide angle camera used in the Venus fly-by is about 74.0 arcsec (10.5° FOV with imaged binned to 512×512), while the minimum angular width of a single pixel for the camera used in the lunar return is about 32.3 arcsec (9.2° FOV with 1024×1024 image).

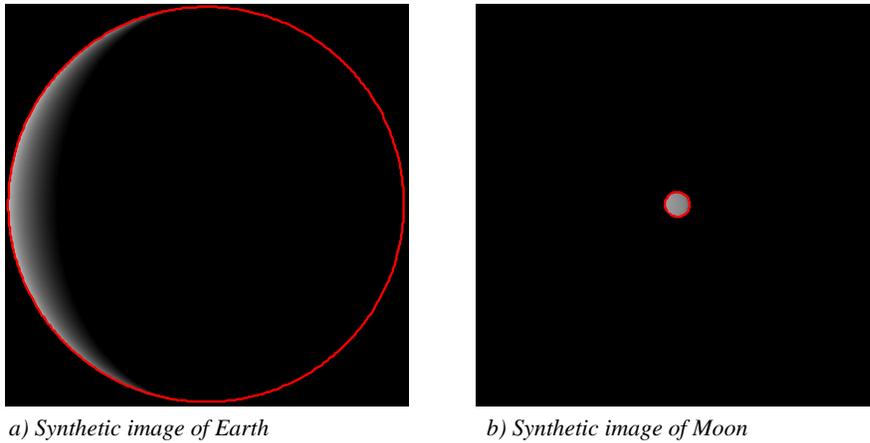
Star horizon measurements are also investigated in this case study. Initial studies demonstrated that it was difficult to obtain reasonable star-horizon



a) Synthetic image of Earth

b) Synthetic image of Moon

Figure 6.9: Synthetic images of the Earth and Moon (with no surface texture map) at point *a* (ID 1) on the reference lunar return trajectory from Fig. 6.8. The raw images are superimposed with red ellipses indicating the horizon location estimated by the image processing algorithm.



a) Synthetic image of Earth

b) Synthetic image of Moon

Figure 6.10: Synthetic images of the Earth and Moon (with no surface texture map) at point *b* (ID 22) on the reference lunar return trajectory from Fig. 6.8. The raw images are superimposed with red ellipses indicating the horizon location estimated by the image processing algorithm.

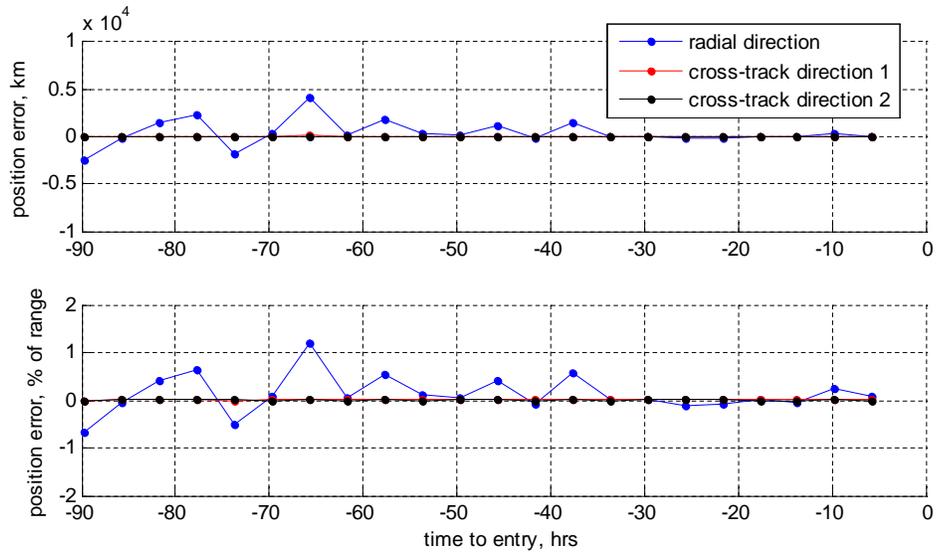


Figure 6.11: Measurement error from instantaneous position fix generated from apparent diameter and centroid of the Earth.

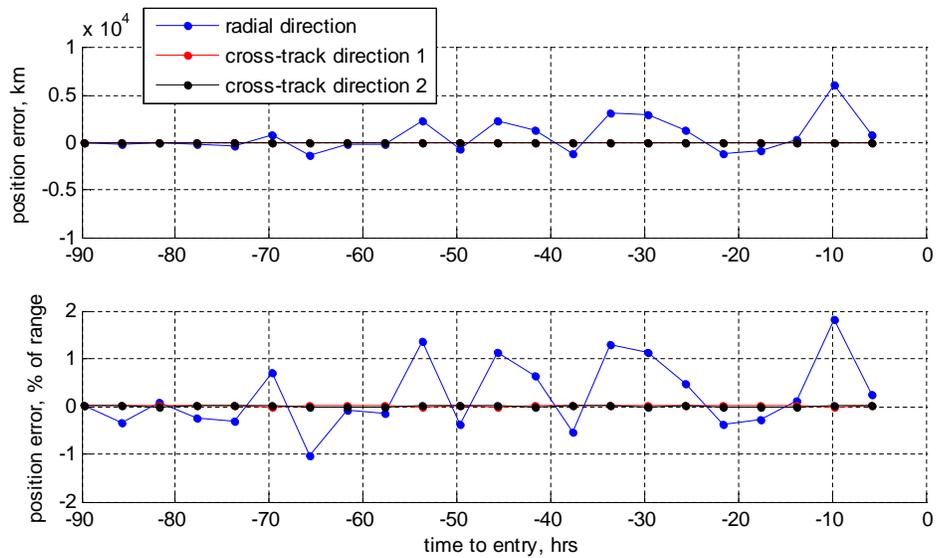


Figure 6.12: Measurement error from instantaneous position fix generated from apparent diameter and centroid of the Moon.

measurements in a single image. If the image had a short exposure time to obtain a good planet horizon measurement, only very bright stars would be visible. Stars of sufficient brightness were too few in number to be practically useful. Even if these bright stars were detected, the signal-to-noise ratio was much lower than desired. Alternatively, if the exposure time was lengthened, then the planet becomes overexposed. This creates problems in accurately determining the horizon location. Therefore, in this study, information from two separate images is combined to generate star-horizon measurements. One image with a short exposure time is used to create the horizon estimate. Another image with a longer exposure time is used to detect stars.

An additional factor contributing to improved performance of the lunar return over the Venus fly-by is the direction of spacecraft motion. In the Venus fly-by, the spacecraft was moving away from the planet, causing both the propagated *a-priori* state covariance and the measurement covariance to grow (especially in the radial direction) with each successive image. For the case of a lunar return, however, the spacecraft is approaching the Earth. Although the propagated *a-priori* state covariance still grows between images, the measurement covariance associated with an image that contain the Earth becomes smaller with each each successive image. This creates a stabilizing effect on the filtered state estimate that prevents the continued covariance growth in the radial direction that was seen in the Venus fly-by example.

Two different kinds of optical navigation measurements are available for inclusion into the EKF. If only centroid and apparent diameter measurements

of both the Earth and Moon are used, a representative example of the filtered position and velocity error is shown in Fig. 6.13 and Fig. 6.14, respectively. If, on the other hand, only star-horizon measurements are used (again using both the Earth and the Moon), the filtered results are as shown Fig. 6.15 and Fig. 6.16. If both measurement types are used, the results are as shown Fig. 6.17 and Fig. 6.18. The example shown in here assumes apparent diameter and centroid measurements of the Earth only and star-horizon measurements from the Moon only. It was found that when both measurement types are used, it is better to only use one measurement type from each image at each time. The error in apparent diameter and centroid measurement and the star-horizon measurement are correlated (because both of these measurement types rely on the best-fit ellipse of the horizon). Including two measurements from the same image in an EKF update causes an unrealistic reduction in the covariance that can lead to filter divergence. Increasing process noise may prevent filter divergence in this case, but introduces other problems. Therefore all cases shown here use one Earth-based measurement and one Moon-based measurement at each measurement time.

If the estimated state and covariance at each time are propagated forward to entry interface, the uncertainty in the entry FPA may be assessed. This quantity is important in an autonomous lunar return because it is the current onboard estimate of the entry FPA that will be used to determine if a trajectory correction maneuver is required. The estimated entry FPA errors for the three measurement scenarios described above are shown in Fig. 6.19,

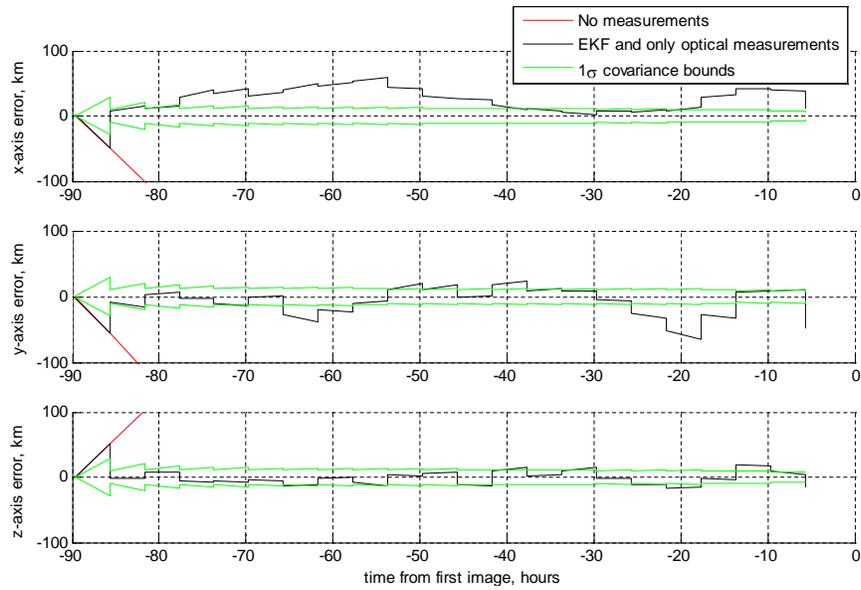


Figure 6.13: Filtered position error for autonomous lunar return using only apparent diameter and centroid measurements of the Earth and Moon.

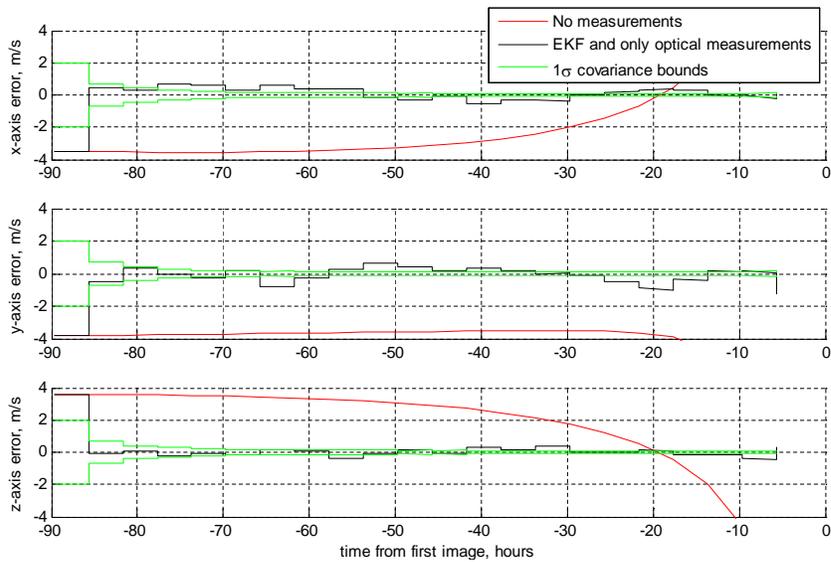


Figure 6.14: Filtered velocity error for autonomous lunar return using only apparent diameter and centroid measurements of the Earth and Moon.

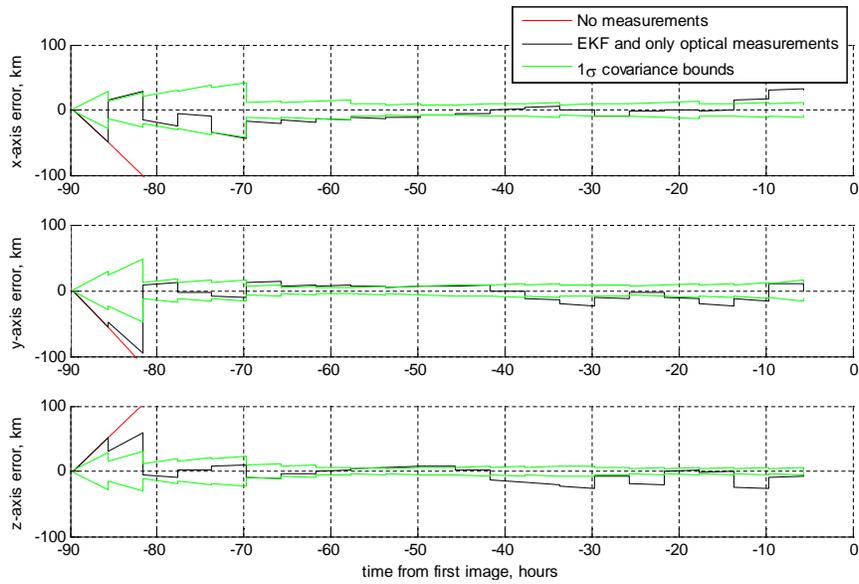


Figure 6.15: Filtered position error for autonomous lunar return using only star-horizon measurements measured from the Earth and Moon.

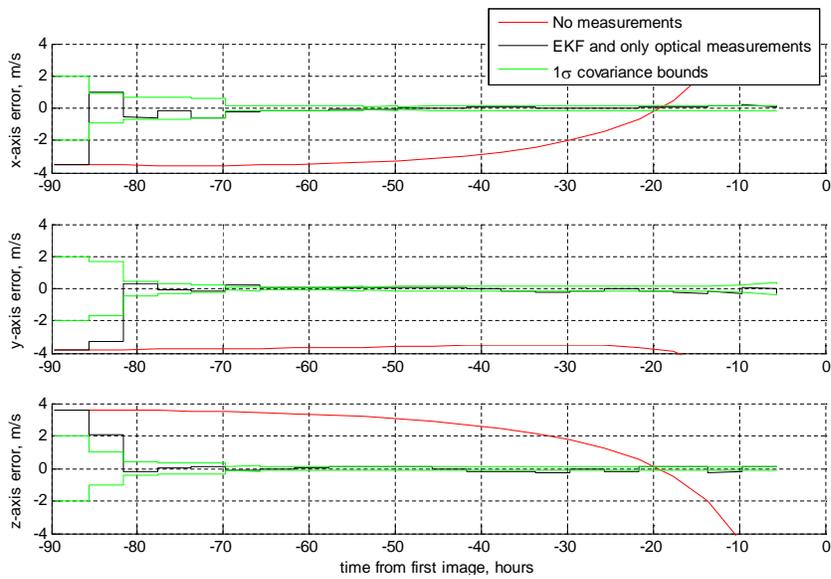


Figure 6.16: Filtered velocity error for autonomous lunar return using only star-horizon measurements measured from the Earth and Moon.

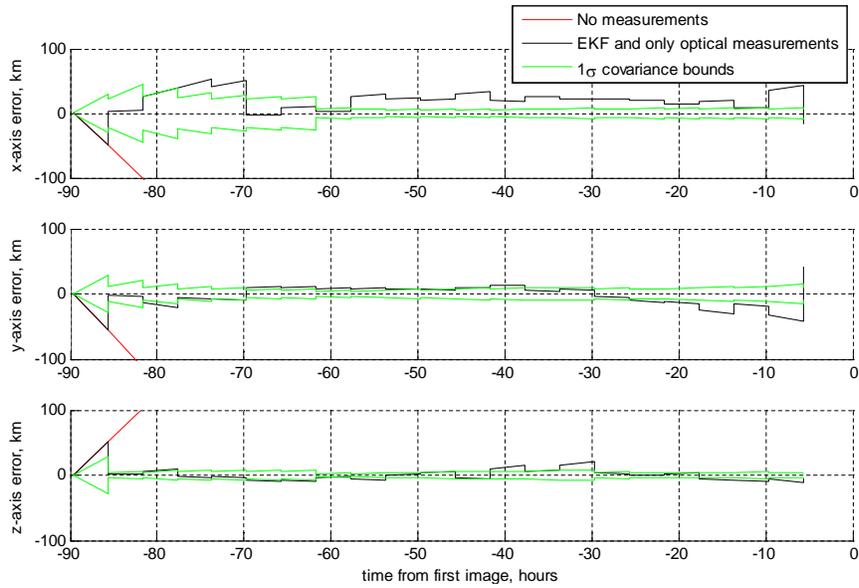


Figure 6.17: Filtered position error for autonomous lunar return using apparent diameter, centroid, and star-horizon measurements.

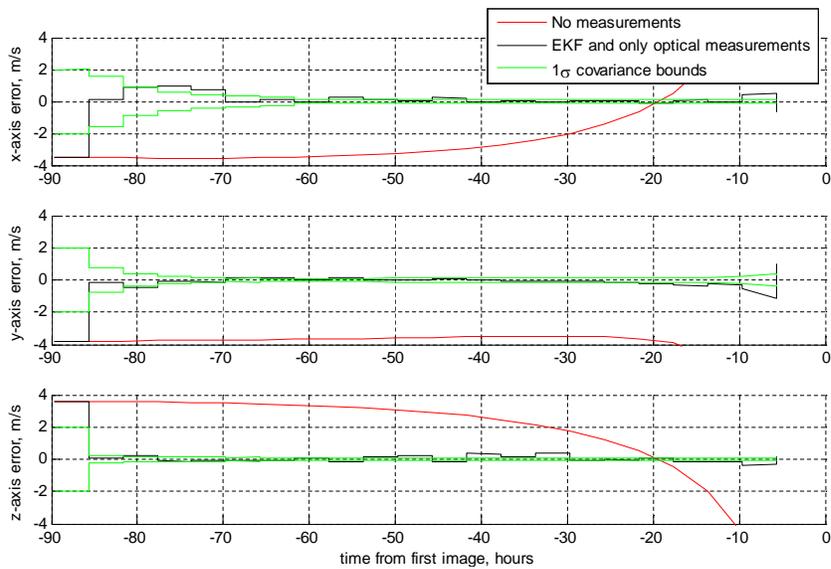


Figure 6.18: Filtered velocity error for autonomous lunar return using apparent diameter, centroid, and star-horizon measurements.

Fig. 6.20, and Fig. 6.21.

All three measurement scenarios are capable of meeting the $\pm 0.5^\circ$ entry FPA error constraint. Further, the 1σ covariance for the predicted entry FPA falls below the entry FPA error constraint more than 24 hours prior to entry for all cases. The apparent diameter and centroid only measurement scenario (Fig. 6.19) appears to provide the quickest covariance convergence. This is likely because the centroid and apparent diameter measurements provide an estimate of the range to the planet (although the covariance of the measurement is very large in the radial direction) and the star-horizon measurements do not provide any range information.

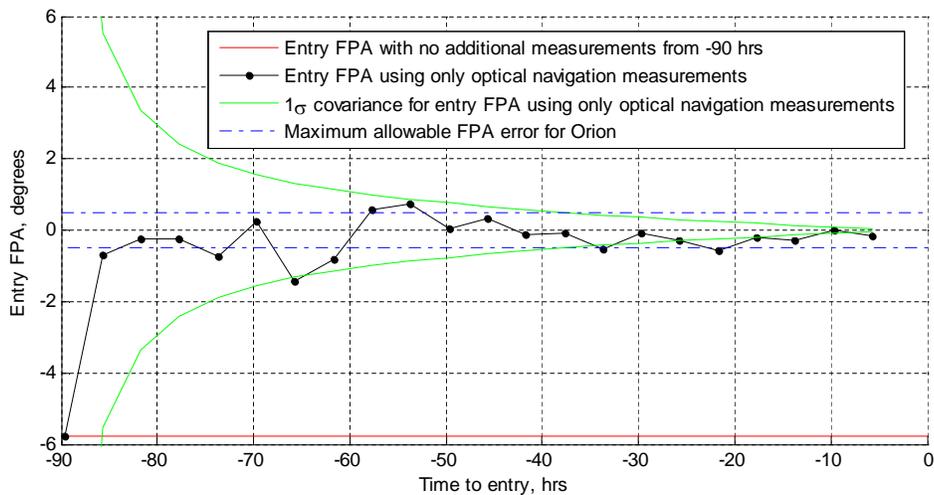


Figure 6.19: Entry FPA error generated by propagating current estimate state and covariance forward to entry interface. Results are for filter that only uses apparent diameter and centroid measurements of the Earth and Moon.

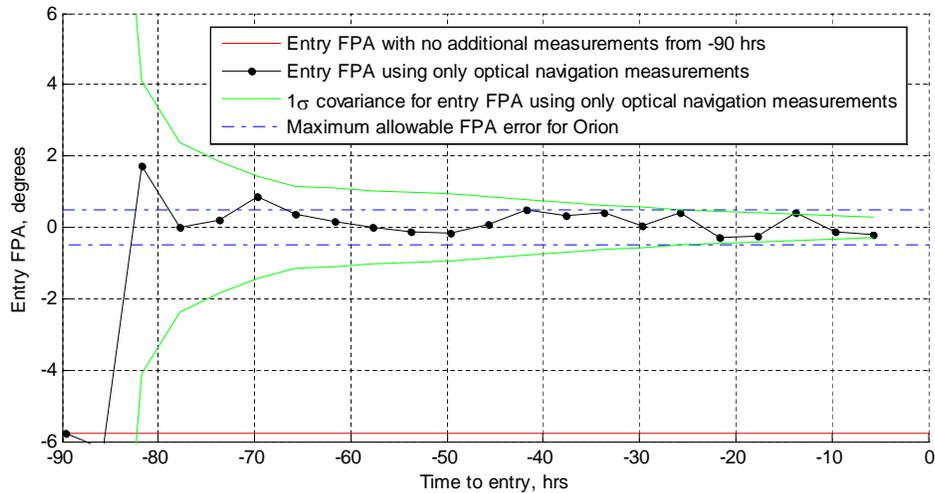


Figure 6.20: Entry FPA error generated by propagating current estimate state and covariance forward to entry interface. Results are for filter that only uses star-horizon measurements measured from the Earth and Moon.

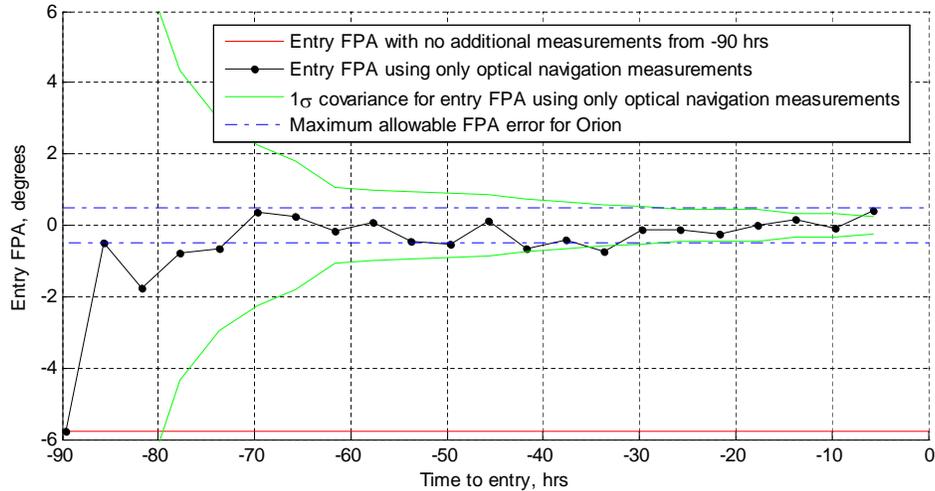


Figure 6.21: Entry FPA error generated by propagating current estimate state and covariance forward to entry interface. Results are for filter that uses apparent diameter and centroid measurements of the Earth only, and star-horizon measurements from the Moon only.

Chapter 7

Technology Development

Engineering is an applied field. Therefore, the work of the previous chapters would be incomplete without a serious discussion of what is necessary to take these ideas from paper to reality.

Beyond the theoretical contributions of this dissertation that allow for an improved assessment of the performance of an optical navigation system (see Section 1.5 for a full discussion of the contributions of this work), a few of these contributions represent *technological* advances that directly lend themselves to implementation on a functioning spacecraft. The new technology here is not the optical measurement types - the idea of using planet apparent diameter, planet centroid, and star-horizon measurements for spacecraft navigation is well established and has been in use since the 1960s. Rather, the new technology introduced in this work is in how these measurement types are obtained from an image. Further, and perhaps most significant, is the introduction of a new attitude filtering method.

Using the methods outlined in the NASA Systems Engineering Handbook, [189] technology assessment is a two-step process. The first step is determining the current Technology Readiness Level (TRL). A TRL assess-

ment, as performed in Section 7.1, helps establish the current maturity of the technology in question. Once the current state of the technology is known, the second step is to conduct an Advancement Degree of Difficulty (AD²) assessment to determine issues associated with maturing the technology from the current state-of-the-art to what is necessary for implementation in a flight system. This analysis is performed in Section 7.2.

7.1 Technology Readiness Level

The 1-9 TRL scale shown in Fig. 7.1 was first developed by NASA in the 1980s. Detailed discussions of this system and its applications may be found in [189] and [190]. The use of this TRL scale to evaluate the maturity of technologies is commonplace in NASA [189] and the Department of Defense.[191] The TRL scale has been shown to provide a useful framework for tracking the progress of technology development over a wide array of different applications.

Both the image processing algorithm for extracting optical navigation measurements and the SOAR Filter are believed to be currently at TRL 3. The exit criteria that NASA uses to determine if TRL 3 has been completed is “documented analytical/experimental results validating predictions of key parameters.”[190] The exit criteria for TRL 4, on the other hand, requires “documented test performance demonstrating agreement with analytical predictions.”[190]

For the image processing algorithm, the work in Chapter 4 and Chapter

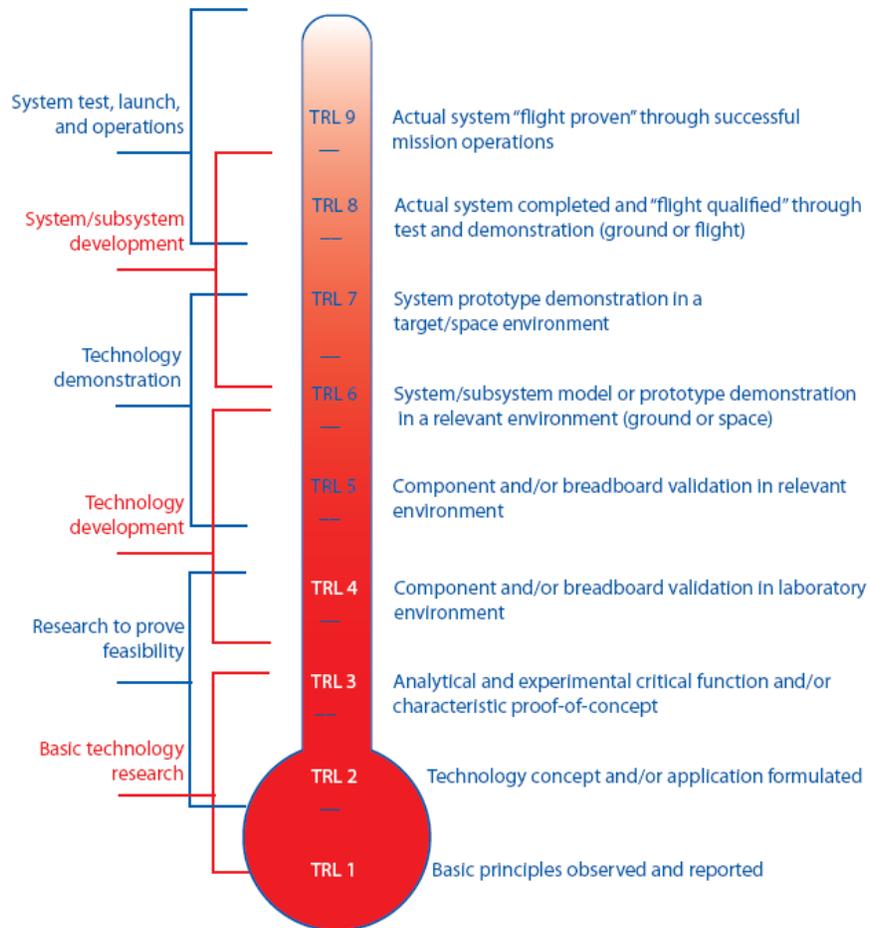


Figure 7.1: The Technology Readiness Level (TRL) scale. Original image from [189], courtesy of NASA.

6 of this dissertation clearly satisfies the criteria for completing TRL 3. The Venus fly-by case study demonstrates the functionality of the image processing algorithm (as written in MATLAB) using real mission data. The lunar return case study provides another demonstration of the algorithm's performance

using simulated images. These results indicate that this image processing algorithm is capable of providing the predicted optical navigation information. To complete TRL 4, however, requires a breadboard validation in a laboratory environment. The work of this dissertation stopped short of performing the laboratory-based hardware demonstrations that would be required to advance this technology to TRL 4.

For the SOAR Filter, the work in Chapter 5 of this dissertation also satisfies the criteria for completing TRL 3. Because this is a completely new algorithm, this dissertation provides all that is necessary to complete TRL 1-3. This discussion of the SOAR Filter contains the initial theoretical foundations necessary for TRL 1 and the algorithm development (including feasibility and benefits) necessary for TRL 2. The simulations at the end of Chapter 5 and in the second case study of Chapter 6 validate the functionality and performance of the SOAR Filter (as written in MATLAB) using simulated measurements. These results indicate that the SOAR Filter is capable of estimating attitude and non-attitude states with a performance that exceeds that of existing filters in many scenarios. As with the image processing algorithm, laboratory validation of this filter was not performed. The SOAR Filter, therefore, has not yet been matured to a TRL 4.

7.2 Advancement Degree of Difficulty

Given that these technologies are still at a low TRL, it is worthwhile to consider how they could be matured to the point that missions may consider

implementing them. Because most NASA missions require a technology to be at TRL 6 by the Preliminary Design Review (PDR), focus will be placed on advancing these technologies from their current state of TRL 3 through TRL 6. Note, however, that TRL 6 cannot be reached without a detailed design of the entire technology subsystem, which may require the advancement from TRL 5 to TRL 6 to be targeted towards a particular mission.

7.2.1 Image Processing Algorithm

A summary of the recommended steps for advancing the image processing algorithm to TRL 6 is shown in Table 7.1.

The first major challenge with obtaining TRL 6 for this image processing algorithm is testing in a “relative environment.” For TRL 5, it is expected that a scaled relevant environment may be constructed. Depending on testbed availability, mission needs, and the available budget, there are numerous methods for doing this. The first is to have a static test article viewing a static object, such as an illuminated sphere. Images may then be taken under varying ambient lighting conditions and with the target object being illuminated from different directions (and with varying intensity). This process may be repeated under many different static configurations. A more comprehensive (and much more complicated and expensive) approach would be to perform this type of analysis on air bearing testbed, similar to the Formation Control Testbed (FCT) at JPL.[192] This testbed is a five degree-of-freedom system with two translational degrees-of-freedom and three rotational degrees of free-

Table 7.1: Summary of recommended steps for maturing the optical navigation imaging processing algorithm from its current state (TRL 3) through TRL 6.

TRL	Recommended steps to complete each TRL
4	Transfer algorithm from MATLAB to C (or a similar language that can easily be run on a real system). Load the software onto a processor on a table-top system, sometimes called a “flat-sat.” Demonstrate operation of the system using simulated measurements. Next, connect the processor (and image processing algorithm) to real sensors. Provide controlled stimuli to the sensors in a laboratory setting and use the results to verify sensor response and the image processing algorithm’s ability to interpret the data from the real sensors. Demonstrate that the laboratory results agree with the performance predicted by simulation. The final definition of the “relevant environment” must also be established.
5	Provide initial demonstration of technology in a relevant environment. Use a camera to take images of an object in the laboratory, such as an illuminated sphere, and demonstrate the autonomous estimation of relative position under many different lighting and noise conditions. This relative positioning exercise could probably be considered a scaled “relevant environment.” Again, demonstrate that the test results agree with the performance predicted by simulation.
6	Demonstrate a prototype of the technology in a relevant environment. For this system to be demonstrated without scaling, images must be acquired from space. Lighting effects associated with the limb of planets with an atmosphere would also be difficult to simulate on the ground. This test must also demonstrate the proper interfaces with other relevant spacecraft subsystems.

dom. This allows for the testing of a system in motion, rather than just a static case. This would also allow for the optical navigation measurements to be fed into a filter, where its information could be combined with real accelerometer and gyro measurements. In addition to the illuminated target object used to

simulate a celestial body, the cameras on the test article could take images of pseudo-stars mounted on a dome around the testbed.

This scaled environment, however, may not be suitable for the demonstration of TRL 6. Completion of TRL 6 frequently requires demonstration of the technology through the full range of expected operating conditions. This cannot be done for optical navigation on the ground without scaling. Further, there are numerous significant environmental effects that may not be easily simulated in ground-based tests. The most important is the affect of a planetary atmosphere, as viewed from space, on the quality of the optical navigation measurements. For this reason, it is anticipated that advancement of the image processing algorithm to TRL 6 will require some form of space-based testing.

The ISS may provide an accessible testbed for demonstrating the image processing algorithm in a relevant environment suitable for advancement to TRL 6. This is reminiscent of the optical navigation experiments that were flown on Skylab in the 1970s, as was discussed in Section 2.2.3 (although the technology being tested here is quite different). Another option is to use data collected by spacecraft currently in orbit, similar to what was done with the Venus fly-by example in Chapter 6. The difference is that an analysis at this level would require access to all of the raw instrument data (e.g. images, accelerometers, gyros, star trackers, etc.) in the exact form as would be seen by the onboard system - this fidelity of data was not available for the Venus fly-by case study. To obtain appropriate images at a regular enough interval to fully demonstrate the image processing algorithm, it may be necessary to specifically

task an existing spacecraft to collect this data, rather than simply pull existing data that has already been collected. Further, obtaining images under the broad range of environmental conditions required to satisfy TRL 6 may require the tasking of more than one existing spacecraft. It may be easier to fly the image processing algorithm as a technology demonstration experiment on a relevant mission, similar in concept to what was done with the Optical Navigation Camera experiment that was flown on the Mars Reconnaissance Orbiter.

Demonstrating of a relevant environment also includes properly simulating interfaces with other spacecraft subsystems. The schematic in Fig. 7.2 provides a notional diagram of some of the major interfaces. The identification and simulation of these interfaces is one of the reasons that it is difficult to advance a technology to TRL 6 without the context of a specific mission.

A system such as this has a large number of interfaces. Therefore, as part of the advancement to TRL 4, it will likely be necessary to draft (or obtain for existing sensors or equipment) an Interface Definition Document (IDD) that explicitly outlines the interface specifications for each component. Further, to perform an integrated test, it will also be necessary to create an Interface Requirements Document (IRD) to define the physical and functional interface between the components at each interface. Depending on the sophistication of the test and equipment, this may be all that is required to achieve TRL 4 (because TRL 4 may be demonstrated with a “breadboard” setup that allows for *ad-hoc* component selection). Advancing to TRL 5 (and, subse-

quently, to TRL 6), however, requires testing in a relevant environment. This requires proper interface modeling. At this point, it will become necessary to draft an Interface Control Document (ICD) which provides a physical description of the interface between two components (e.g. drawing, schematic, etc.). Without an ICD, it is impossible to know that the system is being tested with all of the appropriate interface specifications.

The first set of interfaces that merit discussion are those that occur

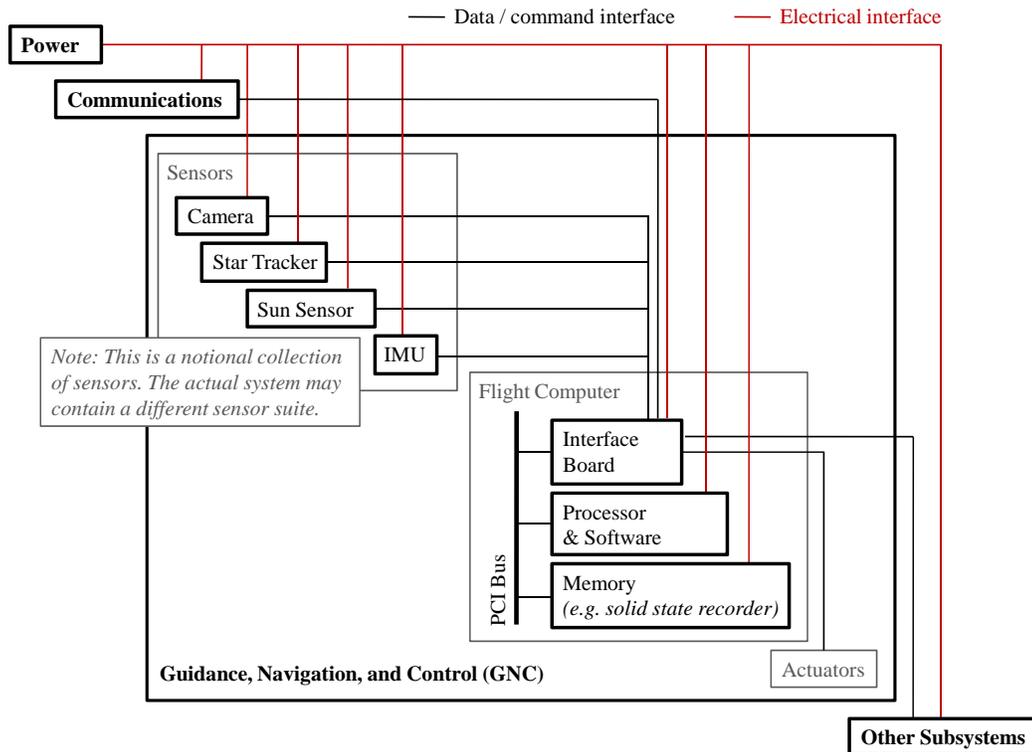


Figure 7.2: Notional schematic of major GNC subsystem interfaces that are important for demonstration of a relevant environment for the image processing algorithm and SOAR Filter. Note that the image processing algorithm and the SOAR Filter are part of the software on the flight computer's processor.

within the GNC subsystem. These interfaces are expected to be the most critical for the creation of a relevant environment. The second class of important interfaces are those that connect components of the GNC subsystem to other spacecraft subsystems. The major connections in this area are expected to be with the power subsystem and with the communications subsystem. Under normal operating conditions, the communication subsystem may be one of the major sources of navigation information (through radiometric tracking). There may also be scenarios where it is desirable to combine optical navigation measurements with radiometric navigation measurements on the spacecraft.

Another major challenge moving forward with the image processing algorithm is expected to be the demonstration of the robustness of the optical navigation measurements. This is especially important for autonomous operations. Further, in the event that the measurement is incorrect, how reliably can the image processing algorithm identify the erroneous measurement? Some simple methods for this type of error checking were presented in Chapter 4. The probability of obtaining a faulty measurement and, subsequently, the probability of detecting that faulty measurement, is expected to be highly dependent on the specific mission scenario and spacecraft hardware. Further, the probabilistic nature of the MSAC algorithm, although it adds robustness to the ellipse model fitting step, makes it harder to prove overall algorithm robustness. It is anticipated that any demonstration or verification of the robustness of this image processing algorithm will be largely empirical (rather than theoretical).

7.2.2 SOAR Filter

A summary of the recommended steps for advancing the SOAR Filter to TRL 6 are shown in Table 7.2.

As with the image processing algorithm, the primary difficulty is testing in a relevant environment. Fortunately, it is easier to obtain a relevant environment for the SOAR Filter. The SOAR Filter only address the problem

Table 7.2: Summary of recommended steps for maturing the SOAR Filter from its current state (TRL 3) through TRL 6.

TRL	Recommended steps to complete each TRL
4	Transfer algorithm from MATLAB to C (or a similar language that can easily be run on a real system). Load the software onto a processor on a table-top system, sometimes called a “flat-sat.” Demonstrate operation of the system using simulated measurements. Next, connect the processor (and SOAR Filter) to real sensors. Provide controlled stimuli to the sensors in a laboratory setting and use the results to verify sensor response and the SOAR Filter’s ability to interpret the data from the real sensors. Demonstrate that the laboratory results agree with the performance predicted by simulation. The final definition of the “relevant environment” must also be established.
5	Provide initial demonstration of the SOAR Filter in a relevant environment. Because this technology deals only with attitude estimation, a relevant environment is simply a platform that freely allows motion in the three rotational degrees-of-freedom. Again, demonstrate that the test results agree with the performance predicted by simulation.
6	Demonstrate a prototype of the technology in a relevant environment. As for TRL 5, the relevant environment is a platform that freely allows motion in the three rotational degrees-of-freedom. This test must also demonstrate the proper interfaces with other relevant spacecraft subsystems.

of attitude filtering. Therefore, any of a number of different testbeds that allow for testing in three rotational degrees-of-freedom environment may be suitable. The method of choice will depend on testbed availability, specific mission needs, and the available budget.

An air bearing testbed may be capable of providing the relevant environment for the demonstrations necessary to achieve TRL 5 and TRL 6. Numerous facilities and institutions have testbeds equipped with a hemispherical air bearing that permits three rotational degrees-of-freedom, such as the Integrated Attitude Control System (IACS) developed at the Georgia Institute of Technology.[193] On such a testbed, the SOAR Filter may be demonstrated using real gyro measurements. Star trackers may take images of pseudo-stars mounted on a dome around the testbed, and the resulting measurements may be used to test the full quaternion measurement input to the SOAR Filter. Realistic simulation of a sun sensor, magnetometer, or horizon sensor should also be performed to demonstrate unit vector measurement inputs to the SOAR Filter.

An alternative test platform is a free floating experiment on a parabolic microgravity flight.¹ The NASA Reduced Gravity Research Program provides a mechanism for researchers to perform short duration tests of equipment in

¹The term “microgravity” is a bit of a misnomer. Gravity, of course, is still there and is acting on the aircraft, test article, and researchers with the same force as if the aircraft were in normal flight. A more accurate description would be to say that the aircraft, and the test article within the aircraft, are in “free fall.” The terms “microgravity” and “reduced gravity” are used here because they are the terminology commonly used to describe this type of environment.

a microgravity environment. This program uses a C-9B aircraft flown out of Ellington Field in Houston, TX, that performs parabolic flights over the Gulf of Mexico. Each parabola provides the researchers onboard the aircraft with approximately 25 seconds of microgravity. There is a precedence for using these parabolic flights to test attitude determination and control systems that require three rotational degrees-of-freedom.[195] As is shown in Fig. 7.3, free floating experiments may be released in the test area during a parabola and are free to translate and rotate in a uninhibited fashion. Although the about 25

Free floating experiments are allowed to translate throughout a specified region of the aircraft cabin and are capable of rotating freely about all three axes (with the exception of a usually negligible amount of drag).

Cabin dimensions and motion of the aircraft typically limit the duration of free floats to 6-8 seconds before the test article impacts the wall of the aircraft or leaves the acceptable test area.

A free floating test article, the Variable Inertia Test Platform (VITP) in this example image, is limited in size – making microgravity flights a good method for demonstrating a technology in flight-relevant form factor.



Figure 7.3: Image of author deploying a three degree-of-freedom attitude control experiment on a parabolic microgravity flight in April 2003. A similar experimental set-up could be used to test the SOAR Filter. The image on the right is jsc2003e27939 from [194], courtesy of NASA.

seconds of microgravity is typically achieved in each parabola, a free floating experiment typically lasts for only 6-8 seconds before its translational motion causes it to impact the wall of the aircraft or exit the test area. Therefore, any testing of the SOAR Filter in this environment would have to work within these time constraints. Tests that require continuous filter operation for longer than a few seconds are not suitable for for this test platform. Alternatively, tests that only require a few seconds of data or tests that would allow multiple short data sets to be combined are good candidates for microgravity flights.

A third potential test platform is a university-built nanosatellite or picosatellite. The University of Texas at Austin, for example, has a vibrant student-built satellite program. The school's Satellite Design Lab (SDL) has a proven track record of designing, building, and delivering small spacecraft.[196] Recent accomplishments of the SDL include PARADIGM, which was deployed from the Space Shuttle Endeavor (STS-127) on 30 July 2009 (see Fig. 7.4), and the larger FASTRAC Spacecraft which is currently scheduled to launch as a payload aboard a Minotaur IV launch vehicle from Kodiak, AK, in May 2010. Small, student-built satellites such as these provide an excellent platform for demonstrating certain types of new technologies - such as the SOAR Filter. Demonstration of this attitude filter on one of these vehicles would be a major step forward in advancing this technology to the point where it may be considered for a larger flight project.

A fourth testing option would be to obtain raw instrument data (e.g. gyros, star trackers, sun sensors, etc.) in the exact form and at the same

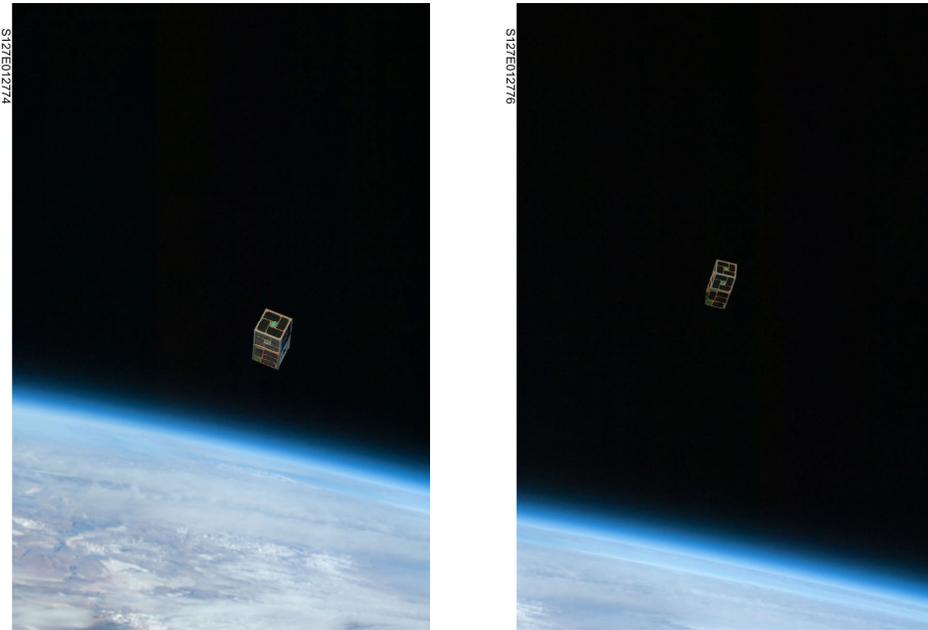


Figure 7.4: Images of PARADIGM deployment taken from the Space Shuttle Endeavor (STS-127) on 30 July 2009. The image on the left is S127-E-012774 and image on the right is S127-E-012776; both images are from [197], courtesy of NASA.

frequency as would be seen by the onboard system. This data could then be fed to the processor across properly simulated instrument interfaces. As with the similar option for testing the optical navigation image processing algorithm, this may require specific tasking of an existing spacecraft. The data needed for this test, however, is much easier to obtain, may be collected from a wider array of existing satellites, and is less intrusive to the day-to-day operations of the existing satellite.

It is worth noting again testing the SOAR Filter in a relevant envi-

ronment requires simulation of the interfaces in addition to the operational conditions. The major interfaces for the SOAR Filter are similar to those for the image processing algorithm, and are shown graphically in Fig. 7.2. The primary difference will be in the sensor suite. The interface requirements that are necessary for advancing through TRL 4, TRL 5, and TRL 6 for the image processing algorithm are also necessary for advancing the SOAR Filter through these same milestones.

For all the SOAR Filter tests and analyses described in Table 7.2, the performance of the SOAR Filter should be compared with the performance of the MEKF (the current state-of-the-art attitude filter).

Chapter 8

Conclusions

8.1 Conclusions

This dissertation provides an in-depth study of optical navigation for a spacecraft in a planetary system. Significant contributions are made in the following four major areas:

1. Optical sensor models
2. Image processing algorithm for extraction of optical navigation measurements
3. New non-linear attitude filter (SOAR Filter)
4. Optical navigation performance analysis

These four items are discussed in more detail in the following paragraphs.

First, in Chapter 3, a new optical sensor model is developed. A detailed discussion of the various error sources (both internal and external to the sensor) provides the foundation for a derivation of the resulting errors in optical line-of-sight measurements. These errors are shown to correspond to specific terms frequently used for camera calibration. A method for creating synthetic images of stars and planets is also introduced. These developments allow for the

construction of a new and more sophisticated representation of the line-of-sight covariance matrix that may be constructed directly from the camera error statistics. These covariance results are compared with other models from the literature.

Second, in Chapter 4, the four standard optical navigation measurement types are introduced. Next, a new image processing algorithm is developed that allows for the robust and autonomous extraction of two types of optical navigation measurements from a raw image. Practical implementation issues are discussed in detail throughout this chapter. The performance of this algorithm is demonstrated on real images from numerous different spacecraft. Additionally, a new theoretical derivation of the measurement covariance is provided and the results are validated through a Monte Carlo analysis using synthetic images. Finally, the measurement sensitivity matrices are derived.

Third, in Chapter 5, a new nonlinear attitude filter called the Sequential Optimal Attitude Recursion (SOAR) Filter is developed. This filter is based on maximum likelihood estimation and is capable of estimating both attitude and non-attitude states. The SOAR Filter provides superior performance (relative to existing attitude filters) when angular errors are large and is shown to provide identical performance to the MEKF when errors are small. The SOAR Filter algorithm is compared to numerous other attitude filtering techniques from a theoretical standpoint. An example simulation is performed where the filter is required to estimate both attitude and gyro bias. In this example, the SOAR Filter provides better performance than the MEKF and Filter QUEST.

Fourth, in Chapter 6, an end-to-end demonstration of completely autonomous optical navigation is provided for two different scenarios: a Venus fly-by and a lunar return. For the Venus fly-by, real images collected by the MESSENGER spacecraft during its 2007 Venus fly-by are used. Optical navigation measurements are autonomously extracted from these images using the image processing algorithms from Chapter 4. This demonstrates that the algorithms developed here work with real images. Post fly-by position and velocity errors are shown to be substantially reduced by the inclusion of optical navigation measurements. Next, for the lunar return, synthetic images of the Earth and Moon are created using the methods from Chapter 3 and then processed using the same image processing algorithms as used in the Venus fly-by example. Performance is shown using different combinations of optical measurements. The requirement for this scenario is to meet a 0.5 degree entry FPA constraint - the predicted 1σ entry FPA error falls within this constraint at about 25-35 hours prior to entry, depending on the measurement combination used.

Additionally, initial recommendations for technology development are outlined in Chapter 7. This chapter highlights that the work of this dissertation advances two different technologies, (1) a new image processing algorithm for the autonomous generation of optical navigation measurements and (2) a new nonlinear attitude filter called the SOAR Filter, from concept generation through TRL 3. The contents of Chapter 7 describe a number of alternative methods for advancing these technologies from their current state through

TRL 6.

8.2 Future Work

There are numerous opportunities to continue the research presented in this dissertation. A sampling of some of these opportunities is provided below.

With regards to the sensor model and synthetic image generation, there are a few different areas that provide opportunities for future work. The first place for future work is the explicit inclusion of a blooming model for CCDs. Although blooming is not a problem in CMOS detectors and only occurs in CCDs when pixels approach (or exceed) their full well capacity, it would be a nice feature to include for the study of over-exposed images. This addition to the CCD detector model could also look at the effect of anti-bloom drains on the sensor response. A second area for future work is the incorporation of albedo surface maps and surface texture maps for planets and moons. The models introduced in Chapter 3 assume that the observed body is a smooth triaxial ellipsoid with a uniform albedo. While this is sufficient for testing the horizon-finding algorithms used here, the inclusion of these surface maps will add an additional layer of realism to the images. A third area of future work is improving the fidelity of the lighting profile models - especially for planets with an atmosphere. In this work, a simple bidirectional reflectance model was used. Although this model may be sufficient for many mission scenarios, it may not be adequate (depending on mission requirements) for spacecraft

operating very close to a planet with an atmosphere, such as spacecraft in LEO. It may also be interesting to look at modeling the light reflected by a planet with Minnaert functions.[112,198]

There is also room for future work on the image processing algorithm. The first opportunity for future work is adding the capability to autonomously determine the lighting direction in the image. The method used in this dissertation assumes that the lighting direction is known from an *a-priori* state estimate or from a sun sensor. It is possible that the lighting direction may be extracted directly from the image without these other sources of information - some references are given in Chapter 4 that show this being done in different applications. Incorporating this capability may help the image processing algorithm be less reliant on other sources of data. The second area for future work is in the method used for improving the location of the horizon points. The current horizon refinement technique uses a series of horizon sweeps and a brute-force method for minimizing the lighting profile residual. This approach is computationally expensive and is significantly less reliable than the remaining components of the image processing algorithm. A better horizon refinement process could improve algorithm performance. In many ways, this improvement goes hand-in-hand with the improvements in the planet lighting profile discussed above. A third area for future work is in developing rigorous error detection algorithms for the image processing algorithm. It will be necessary to develop robust methods for autonomously detecting erroneous measurements if this type of system to be used in a truly autonomous fashion.

Finally, there are numerous opportunities for future work on the SOAR Filter. The first area of future work is taking a formal look at the convergence properties of the SOAR Filter. The examples cases generated as part of this dissertation show empirically that the SOAR Filter generally exhibits superior convergence when compared to other attitude filter types. It would be interesting to see if this could be proved theoretically. A second area of future work is considering how additional measurement types (things other than attitude measurements and unit vector measurements) could be included in the SOAR Filter. A third area is to look at the structure of the state update and covariance update to see if these expressions may be written in a way that maximizes computational speed and algorithm robustness (e.g. minimize use of inverses) for implementation on a real system.

Appendices

Appendix A

Quaternion Mathematics

This appendix describes key results from Quaternion Mathematics - especially as they relate to rotations. The quaternion operations are introduced here for easy reference (and with a nomenclature consistent with the rest of this dissertation) without derivation. More information about quaternion mathematics may be found in [80] and [78].

A.1 Definition of the Quaternion

The quaternion is described by a 4-tuple of real numbers. It consists of a vector component that exists in \mathbb{R}^3 and a scalar component. The nomenclature used for this varies significantly in the literature, but the following notation is used throughout this dissertation

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (\text{A.1})$$

A.2 Quaternion Multiplication

Given two quaternions, $\bar{\mathbf{q}}$ and $\bar{\mathbf{p}}$, define the multiplication of these two quaternions (also called the quaternion product) as

$$\bar{\mathbf{q}} \otimes \bar{\mathbf{p}} = \begin{bmatrix} q_4 \mathbf{p} + p_4 \mathbf{q} - \mathbf{q} \times \mathbf{p} \\ q_4 p_4 - \mathbf{q}^T \mathbf{p} \end{bmatrix} \quad (\text{A.2})$$

where the symbol \otimes denotes the quaternion multiplication operator. Note that this quaternion product is backwards when compared to the classical quaternion product, as is standard practice in the attitude determination community. This is done so that the order of quaternion multiplication mirrors the order of DCM multiplication for sequential rotations. If the classical quaternion product is denoted by \odot , then $\bar{\mathbf{q}} \otimes \bar{\mathbf{p}} = \bar{\mathbf{p}} \odot \bar{\mathbf{q}}$. The only difference in Eq. A.2 for these two conventions is in the sign of the cross product term.

The quaternion product as shown in Eq. A.2 allows for sequential rotations in the same manner as DCMs:

$$\mathbf{T}_C^A = \mathbf{T}_C^B \mathbf{T}_B^A = \mathbf{T}(\bar{\mathbf{q}}_C^B) \mathbf{T}(\bar{\mathbf{q}}_B^A) = \mathbf{T}(\bar{\mathbf{q}}_C^B \otimes \bar{\mathbf{q}}_B^A) = \mathbf{T}(\bar{\mathbf{q}}_C^A) \quad (\text{A.3})$$

Further, it is sometimes convenient to write the quaternion product as a standard matrix-vector multiplication. Rewriting Eq. A.2 in this form will show that

$$\bar{\mathbf{q}} \otimes \bar{\mathbf{p}} = \begin{bmatrix} q_4 \mathbf{I}_{3 \times 3} - [\mathbf{q} \times] & \mathbf{q} \\ -\mathbf{q}^T & q_4 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ p_4 \end{bmatrix} = \Xi(\bar{\mathbf{q}}) \bar{\mathbf{p}} \quad (\text{A.4})$$

A.3 Complex Conjugate and Inverse of a Quaternion

The complex conjugate of the quaternion $\bar{\mathbf{q}}$ is given by

$$\bar{\mathbf{q}}^* = \begin{bmatrix} -\mathbf{q} \\ q_4 \end{bmatrix} \quad (\text{A.5})$$

Next, in keeping with the standard definition of an inverse,

$$\bar{\mathbf{q}} \otimes \bar{\mathbf{q}}^{-1} = \bar{\mathbf{q}}^{-1} \otimes \bar{\mathbf{q}} = 1 \quad (\text{A.6})$$

From this relation, it may be shown that the inverse of a quaternion may be computed from the conjugate as

$$\bar{\mathbf{q}}^{-1} = \frac{\bar{\mathbf{q}}^*}{\|\bar{\mathbf{q}}\|} \quad (\text{A.7})$$

where $\|\bar{\mathbf{q}}\|$ is the norm of the quaternion. If the quaternion in question is an attitude quaternion, then it has unity norm, $\|\bar{\mathbf{q}}\| = 1$. This means that, for an attitude quaternion, the complex conjugate of the quaternion is the same as the inverse of the quaternion. Further, if the attitude quaternion $\bar{\mathbf{q}}$ describes a rotation from frame A to frame B , then it is shown in [80] that the equivalence of the attitude quaternion conjugate and inverse is analogous to the following well known relation for rotation matrices

$$(\mathbf{T}_B^A)^{-1} = (\mathbf{T}_B^A)^T \quad (\text{A.8})$$

A.4 Norm of a Quaternion

The norm of a quaternion is a description of its length. Therefore, define the quaternion norm as

$$\|\bar{\mathbf{q}}\| = \sqrt{\bar{\mathbf{q}}^* \otimes \bar{\mathbf{q}}} \quad (\text{A.9})$$

By the definition of the quaternion product and quaternion complex conjugate, the vector part of the quaternion will be exactly zero. All that remains is the scalar component. Therefore, it is also true that

$$\|\bar{\mathbf{q}}\|^2 = \mathbf{q}^T \mathbf{q} + q_4^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 \quad (\text{A.10})$$

Appendix B

Star Catalogs

A star catalog is a database that contains a list of stars along with information about each of those stars (e.g. right ascension, declination, magnitude, etc.). A very nice discussion of the evolution of star charts and star catalogs from antiquity to modern times may be found in [199].

In modern spacecraft applications, existing star catalogs are typically used to generate mission-specific star catalogs for use onboard the vehicle. If information regarding the star brightness is to be used by onboard algorithms, the star magnitudes from the reference catalog must be adjusted according to the spectral sensitivity of the detector. Further, and more importantly, star trackers are typically defocused. This means that two stars that appear close to one another will blur together and appear as a single star, rather than appear as two distinct stars. This phenomena can introduce a significant bias in the measured direction to the observed star. This problem is discussed in more detail in [200] and [19].

There are a number of different star catalogs that are frequently used to create the star catalog stored onboard the spacecraft. Some of these are summarized in Table B.1.

Table B.1: Summary of some of the star catalogs that are commonly used as a starting point for creating mission-specific star catalogs on modern spacecraft.

Star Catalog	Year Created	Number of Stars	Suggested References
SKY2000 V5 Master Catalog	2005/2006	299,460	[201]
HIPPARCOS Catalog	1997	118,218	[202], [203]
Tycho-2 Catalog	2000	2,539,913	[202], [204]

Appendix C

Numerical Algorithm for Computing the Direction to the Center of an Object from a Projected Ellipse

Suppose that an ellipse on a detector is formed by the perspective projection of a sphere or ellipsoid. The line-of-sight direction from the camera to the three dimensional center of the observed object is not formed by a vector originating at the center of the ellipse. In other words, using the ellipse centroid in Eq. 3.22 will not form the line-of-sight direction to the center of the observed object. This fact may be seen graphically in Fig. C.1.

Therefore, an iterative scheme was developed by the author to find the point on the detector plane that creates a line-of-sight to the center of the observed object. The approach derived here will work regardless of if the target object is a sphere or an ellipsoid.

In this iterative scheme, let the estimate of the point on the detector that produces a line-of-sight to the center of the observed object at iteration k be described by $[x_{c,k}, y_{c,k}]$ and let the geometric center of the ellipse generated by the image processing algorithm be described by $[\tilde{x}_0, \tilde{y}_0]$. The value of $[\tilde{x}_0, \tilde{y}_0]$ does not change from iteration to iteration. Initialize this routine by assuming

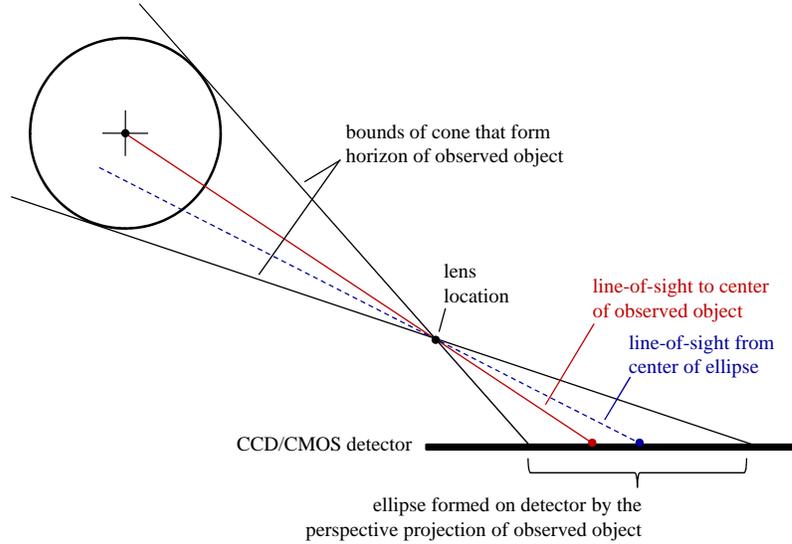


Figure C.1: Graphical depiction of the perspective projection of a sphere onto the detector plane. Note that the three-dimensional center of the observed object does not project to the centroid of the ellipsoid on the image plane. Angles are exaggerated for clarity.

that

$$\begin{bmatrix} x_{c,0} \\ y_{c,0} \end{bmatrix} = \begin{bmatrix} \tilde{x}_0 \\ \tilde{y}_0 \end{bmatrix} \quad (\text{C.1})$$

This initial guess would be true if the line-of-sight to the observed object lies exactly along the camera boresight direction.

Now, use the current guess of the line-of-sight direction to the target object center to compute the range to the target object using Eq. 4.85. Then use the relations of Eq. 4.89 through Eq. 4.93 to generate the ellipse that would be created if this guess were true. If the resulting location of the ellipse center at this iteration, $[x_{0,k}, y_{0,k}]$, does not agree (to within some tolerance) with the measured location of the ellipse center from the image processing

algorithm, $[\tilde{x}_0, \tilde{y}_0]$, then an update is required. This may be performed by a simple update of the form

$$\begin{bmatrix} x_{c,k+1} \\ y_{c,k+1} \end{bmatrix} = \begin{bmatrix} x_{c,k} \\ y_{c,k} \end{bmatrix} + \left\{ \begin{bmatrix} \tilde{x}_0 \\ \tilde{y}_0 \end{bmatrix} - \begin{bmatrix} x_{0,k} \\ y_{0,k} \end{bmatrix} \right\} \quad (\text{C.2})$$

which is to say that the update of $[x_{c,k}, y_{c,k}]$ behaves approximately like the update of $[x_{0,k}, y_{0,k}]$. In practice, this iterative routine was found to be extremely stable. In all cases this routine converged to within machine precision in 3-4 iterations. To date, no cases have been experienced where this iterative routine fails. No formal study of the routine's stability has been performed.

Appendix D

Derivation of The QUaternion ESTimation (QUEST) Algorithm

The QUEST derivation presented here is based on the work of Shuster and Oh.[157] This derivation picks up where Eq. 5.37 leaves off. Recall that the solution to the Wahba problem may be written as

$$\mathbf{K}\bar{\mathbf{q}} = \lambda\bar{\mathbf{q}} \quad (\text{D.1})$$

Here, the 4×4 matrix \mathbf{K} is the Davenport matrix given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} - \mu\mathbf{I}_{3 \times 3} & \mathbf{z} \\ \mathbf{z}^T & \mu \end{bmatrix} \quad (\text{D.2})$$

where

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^T \quad (\text{D.3})$$

$$\mu = \text{tr}[\mathbf{B}] \quad (\text{D.4})$$

$$[\mathbf{z} \times] = \mathbf{B}^T - \mathbf{B} \quad \text{or} \quad \mathbf{z} = \sum_{i=1}^m w_i (\tilde{\mathbf{e}}_i)_B \times (\mathbf{e}_i)_I \quad (\text{D.5})$$

Now, expanding Eq. D.1 using the definitions provided in Eq. 5.32

$$\begin{bmatrix} (\mathbf{S} - \mu\mathbf{I}_{3 \times 3}) \mathbf{q} + \mathbf{z}q_4 \\ \mathbf{z}^T \mathbf{q} + \mu q_4 \end{bmatrix} = \begin{bmatrix} \lambda \mathbf{q} \\ \lambda q_4 \end{bmatrix} \quad (\text{D.6})$$

Further, recalling that the Gibbs vector is given by $\mathbf{g} = \mathbf{q}/q_4$, the above expression may be rearranged to show

$$(\mathbf{S} - \mu \mathbf{I}_{3 \times 3}) \mathbf{g} + \mathbf{z} = \lambda \mathbf{g} \quad (\text{D.7})$$

$$\mathbf{z}^T \mathbf{g} + \mu = \lambda \quad (\text{D.8})$$

Further rearrangement of Eq. D.7 yields

$$\mathbf{g} = [(\lambda + \mu) \mathbf{I}_{3 \times 3} - \mathbf{S}]^{-1} \mathbf{z} \quad (\text{D.9})$$

Proceed by looking for a more convenient way of expressing $[(\lambda + \mu) \mathbf{I}_{3 \times 3} - \mathbf{S}]^{-1}$. Let the eigenvalues of \mathbf{S} be given by ξ :

$$\det [\mathbf{S} - \xi \mathbf{I}_{3 \times 3}] = 0 \quad (\text{D.10})$$

The characteristic equation for an arbitrary 3-by-3 matrix is known to be given by

$$-\xi^3 + 2\sigma\xi^2 - k\xi + \delta = 0 \quad (\text{D.11})$$

where

$$\sigma = \frac{1}{2} \text{tr} [\mathbf{S}] = \text{tr} [\mathbf{B}] = \mu \quad (\text{D.12})$$

$$k = \text{tr} [\text{adj} \mathbf{S}] = -\frac{1}{2} [\text{tr} [\mathbf{S}^2] - \text{tr} [\mathbf{S}]^2] \quad (\text{D.13})$$

$$\delta = \det [\mathbf{S}] \quad (\text{D.14})$$

Finally, by the Cayley-Hamilton theorem[205] \mathbf{S} is known to also satisfy the characteristic equation:

$$-\mathbf{S}^3 + 2\mu\mathbf{S}^2 - k\mathbf{S} + \delta = 0 \quad (\text{D.15})$$

Returning to the problem of rewriting $[(\lambda + \mu) \mathbf{I}_{3 \times 3} - \mathbf{S}]^{-1}$, express this equation as an infinite series in \mathbf{S}

$$f(\mathbf{S}) = [(\lambda + \mu) \mathbf{I}_{3 \times 3} - \mathbf{S}]^{-1} = b_0 \mathbf{I}_{3 \times 3} + \sum_{j=1}^{\infty} b_j \mathbf{S}^j \quad (\text{D.16})$$

Note from Eq. D.15 that any \mathbf{S}^j for $j \geq 3$ may be rewritten as a quadratic function of \mathbf{S} . Therefore, let $f(\mathbf{S})$ be given by the following quadratic function

$$f(\mathbf{S}) = [(\lambda + \mu) \mathbf{I}_{3 \times 3} - \mathbf{S}]^{-1} = \gamma^{-1} [\alpha \mathbf{I}_{3 \times 3} + \beta \mathbf{S} + \mathbf{S}^2] \quad (\text{D.17})$$

Multiplying both sides of the expression by $[(\lambda + \mu) \mathbf{I}_{3 \times 3} - \mathbf{S}]$,

$$\mathbf{I}_{3 \times 3} = \gamma^{-1} [\alpha \mathbf{I}_{3 \times 3} + \beta \mathbf{S} + \mathbf{S}^2] [(\lambda + \mu) \mathbf{I}_{3 \times 3} - \mathbf{S}] \quad (\text{D.18})$$

Expanding this function, eliminating cubic terms of \mathbf{S} with Eq. D.15, and collecting in terms of $\mathbf{I}_{3 \times 3}$, \mathbf{S} , and \mathbf{S}^2 yields

$$\mathbf{I}_{3 \times 3} = \gamma^{-1} [((\lambda + \mu)\alpha - \delta) \mathbf{I}_{3 \times 3} + ((\lambda + \mu)\beta - \alpha + k) \mathbf{S} + ((\lambda + \mu) - \beta - 2\mu) \mathbf{S}^2] \quad (\text{D.19})$$

Recognizing that $\mathbf{I}_{3 \times 3}$, \mathbf{S} , and \mathbf{S}^2 are linearly independent, separate the above equation by coefficients:

$$1 = \gamma^{-1} ((\lambda + \mu)\alpha - \delta) \quad (\text{D.20})$$

$$\gamma = (\lambda + \mu)\alpha - \delta = (\lambda + \mu)(\lambda^2 - \mu^2 + k) - \delta$$

$$0 = (\lambda + \mu)\beta - \alpha + k \quad (\text{D.21})$$

$$\alpha = (\lambda + \mu)\beta + k = \lambda^2 - \mu^2 + k$$

$$0 = (\lambda + \mu) - \beta - 2\mu \quad (\text{D.22})$$

$$\beta = \lambda + \mu - 2\mu = \lambda - \mu$$

Substituting Eq. D.17 into Eq. D.9 an easily obtainable solution to the Gibbs vector in terms of α , β , and γ is given by

$$\mathbf{g} = \gamma^{-1} [\alpha \mathbf{I}_{3 \times 3} + \beta \mathbf{S} + \mathbf{S}^2] \mathbf{z} = \frac{1}{\gamma} \mathbf{y} \quad (\text{D.23})$$

where

$$\mathbf{y} = [\alpha \mathbf{I}_{3 \times 3} + \beta \mathbf{S} + \mathbf{S}^2] \mathbf{z} \quad (\text{D.24})$$

Unfortunately to obtain α , β , and γ (and hence the solution), it is necessary to first find the eigenvalue, λ . By substituting Eq. D.23 into Eq. D.8,

$$\lambda = \mu + \mathbf{z}^T \gamma^{-1} [\alpha \mathbf{I}_{3 \times 3} + \beta \mathbf{S} + \mathbf{S}^2] \mathbf{z} \quad (\text{D.25})$$

After plugging in values for α , β , and γ , it is clear that the above equation is quartic in λ ,

$$p(\lambda) = c_4 \lambda^4 + c_3 \lambda^3 + c_2 \lambda^2 + c_1 \lambda + c_0 = 0 \quad (\text{D.26})$$

Some tedious algebra will show that

$$\begin{aligned} c_4 &= 1 \\ c_3 &= 0 \\ c_2 &= -2\mu^2 + k - \mathbf{z}^T \mathbf{z} \\ c_1 &= -\delta - \mathbf{z}^T \mathbf{S} \mathbf{z} \\ c_0 &= -\mu (k\mu - \delta - \mu^3) - \mathbf{z}^T [(k - \mu^2) \mathbf{I}_{3 \times 3} - \mu \mathbf{S} + \mathbf{S}^2] \mathbf{z} \end{aligned} \quad (\text{D.27})$$

The root in Eq. D.25 may be solved through Newton-Raphson[206] iteration:

$$\frac{dp(\lambda)}{d\lambda} = 4\lambda^3 + 2c_2\lambda + c_1 \quad (\text{D.28})$$

$$\lambda_{k+1} = \lambda_k - \frac{\lambda_k^4 + c_2\lambda_k^2 + c_1\lambda_k + c_0}{4\lambda_k^3 + 2c_2\lambda_k + c_1} \quad (\text{D.29})$$

Because the largest eigenvalue is known to be close to λ_0 , this should be picked as the initial guess for the iteration routine. From here, it is possible to compute α , β , and γ directly with easily obtainable values. Therefore, the attitude quaternion may be computed as

$$\bar{\mathbf{q}} = \frac{1}{\sqrt{1 + \mathbf{g}^T \mathbf{g}}} \begin{bmatrix} \mathbf{g} \\ 1 \end{bmatrix} = \frac{1}{\sqrt{\gamma^2 + \mathbf{y}^T \mathbf{y}}} \begin{bmatrix} \mathbf{y} \\ \gamma \end{bmatrix} \quad (\text{D.30})$$

Note that the solution method used in the QUEST algorithm introduces a singularity for a 180 degree rotation. In practice, this singularity is avoided through the use of successive rotations. If the attitude is found to be near a singularity, a 180 degree rotation is performed about the x-axis and then the QUEST algorithm attempts to find the attitude. If the singularity persists, 180 degree rotations about the y-axis and z-axis are attempted. After the singularity has been removed and the attitude is computed with respect to the rotated frame, the appropriate 180 degree rotation must be undone to arrive at the vehicle attitude with respect to the desired frame.

A flow chart describing the actual implementation of the QUEST algorithm is provided in Fig. D.1.

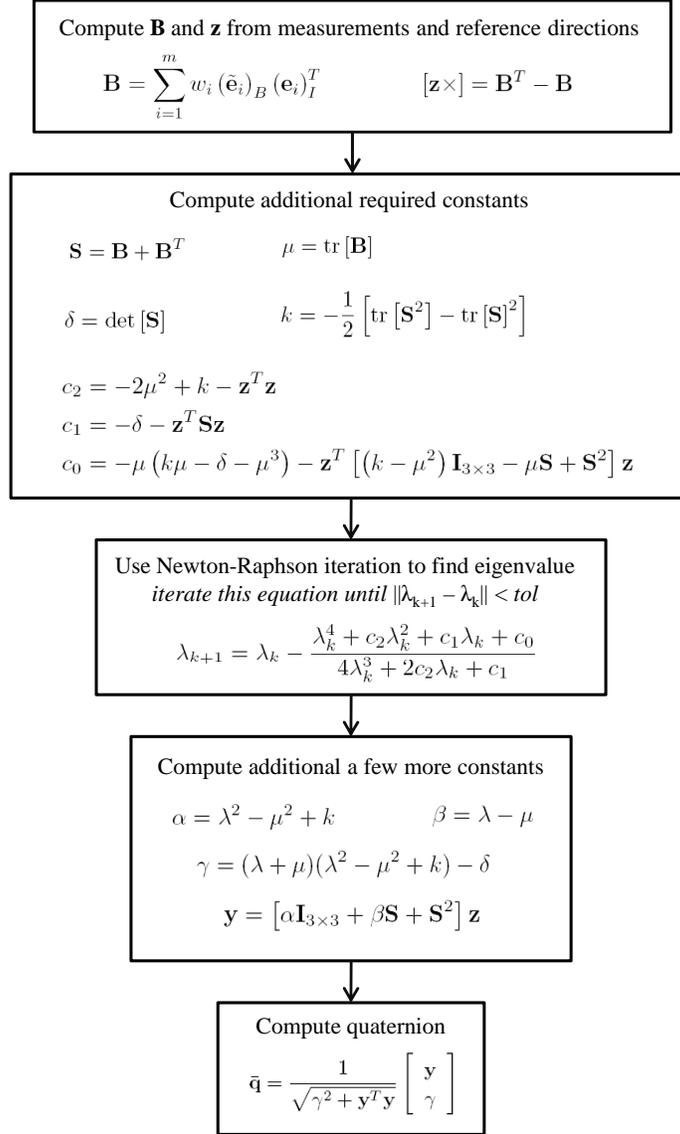


Figure D.1: Flow chart for QUEST algorithm.

Appendix E

Derivation of Relation Between Fisher Information Matrix and Attitude Profile Matrix

In Chapter 5, equations are presented that allow for the transformation between the 3×3 Fisher Information Matrix for the attitude-only problem, $\mathcal{F}_{\theta\theta}$, and the 3×3 attitude profile matrix, \mathbf{B} . These relations, given in Eq. 5.54 and Eq. 5.55, were originally developed by Shuster in [120]. These relations are as follows:

$$\mathbf{P}_{\theta\theta}^{-1} \approx \mathcal{F}_{\theta\theta} = \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \mathbf{I}_{3 \times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (\text{E.1})$$

$$\mathbf{B} = \left[\frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3 \times 3} - \mathcal{F}_{\theta\theta} \right] \mathbf{T}_{\hat{B}}^I \quad (\text{E.2})$$

Although these results are well known and widely used, the details of the derivations are not readily available in the literature. This appendix provides a detailed derivation of both of these results.

E.1 Derivation of $\mathcal{F}_{\theta\theta}$ as function of \mathbf{B}

The objective of this section is to prove the following result

$$\mathcal{F}_{\theta\theta} = \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \mathbf{I}_{3 \times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (\text{E.3})$$

Begin by recalling, from Eq. 5.52, that the Wahba Problem objective function may be written as

$$J(\boldsymbol{\delta\theta}) = \lambda_0 - \text{tr} \left[\left(\mathbf{I}_{3 \times 3} + [-\boldsymbol{\delta\theta} \times] + \frac{1}{2} [-\boldsymbol{\delta\theta} \times]^2 \right) \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \right] \quad (\text{E.4})$$

and that, from Eq. 5.47, the Fisher Information Matrix may be found by

$$\mathcal{F}_{\theta\theta} = E \left[\frac{\partial^2 J(\boldsymbol{\delta\theta})}{\partial \boldsymbol{\delta\theta} \partial \boldsymbol{\delta\theta}} \right] \quad (\text{E.5})$$

Because Eq. E.5 requires the second derivative of $J(\boldsymbol{\delta\theta})$ with respect to $\boldsymbol{\delta\theta}$, terms in $J(\boldsymbol{\delta\theta})$ that are scalar or linear in $\boldsymbol{\delta\theta}$ are unimportant in the computation of $\mathcal{F}_{\theta\theta}$. Therefore,

$$\mathcal{F}_{\theta\theta} = E \left[\frac{\partial^2 J(\boldsymbol{\delta\theta})}{\partial \boldsymbol{\delta\theta} \partial \boldsymbol{\delta\theta}} \right] = E \left[\frac{\partial^2}{\partial \boldsymbol{\delta\theta} \partial \boldsymbol{\delta\theta}} \left(-\text{tr} \left[\frac{1}{2} [-\boldsymbol{\delta\theta} \times]^2 \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \right] \right) \right] \quad (\text{E.6})$$

To compact notation, define the matrix \mathbf{W} as

$$\mathbf{W} = \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (\text{E.7})$$

It is interesting to note that by inserting the definition of \mathbf{B} from Eq. 5.15,

$$\mathbf{W} = \mathbf{T}_{\hat{B}}^I \mathbf{B}^T = \sum_{i=1}^m w_i \mathbf{T}_{\hat{B}}^I (\mathbf{e}_i)_I (\tilde{\mathbf{e}}_i)_B^T = \sum_{i=1}^m w_i (\mathbf{e}_i)_{\hat{B}} (\tilde{\mathbf{e}}_i)_B^T \quad (\text{E.8})$$

it may be observed that \mathbf{W} is symmetric in the noise-free case. Even in the presence of noise, \mathbf{W} should be very nearly symmetric. Therefore, for the present derivation, assume that $\mathbf{W} \approx \mathbf{W}^T$. Substituting \mathbf{W} into Eq. E.6,

$$\mathcal{F}_{\theta\theta} = -\frac{1}{2} E \left[\frac{\partial^2}{\partial \boldsymbol{\delta\theta} \partial \boldsymbol{\delta\theta}} \left(\text{tr} \left[[-\boldsymbol{\delta\theta} \times]^2 \mathbf{W} \right] \right) \right] \quad (\text{E.9})$$

Although it may not be the most elegant approach, term-by-term differentiation is a straightforward and effective method for computing $\mathcal{F}_{\theta\theta}$. The fact that this is only a 3×3 matrix makes this tractable. Begin by expanding the terms within the trace operator. If $\boldsymbol{\delta\theta} = [\delta\theta_1 \ \delta\theta_2 \ \delta\theta_3]^T$, then explicitly expanding $[-\boldsymbol{\delta\theta} \times]^2$ yields

$$[-\boldsymbol{\delta\theta} \times]^2 = \begin{bmatrix} -\delta\theta_2^2 - \delta\theta_3^2 & \delta\theta_1\delta\theta_2 & \delta\theta_1\delta\theta_3 \\ \delta\theta_1\delta\theta_2 & -\delta\theta_1^2 - \delta\theta_3^2 & \delta\theta_2\delta\theta_3 \\ \delta\theta_1\delta\theta_3 & \delta\theta_2\delta\theta_3 & -\delta\theta_1^2 - \delta\theta_2^2 \end{bmatrix} \quad (\text{E.10})$$

Further, if the elements in \mathbf{W} are given by

$$[-\boldsymbol{\delta\theta} \times]^2 = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \quad (\text{E.11})$$

then the diagonal terms of the 3×3 matrix $[-\boldsymbol{\delta\theta} \times]^2 \mathbf{W}$ are as follows

$$[[-\boldsymbol{\delta\theta} \times]^2 \mathbf{W}]_{11} = \delta\theta_1\delta\theta_2w_{21} - w_{11}(\delta\theta_2^2 + \delta\theta_3^2) + \delta\theta_1\delta\theta_3w_{31} \quad (\text{E.12})$$

$$[[-\boldsymbol{\delta\theta} \times]^2 \mathbf{W}]_{22} = \delta\theta_1\delta\theta_2w_{12} - w_{22}(\delta\theta_1^2 + \delta\theta_3^2) + \delta\theta_2\delta\theta_3w_{32} \quad (\text{E.13})$$

$$[[-\boldsymbol{\delta\theta} \times]^2 \mathbf{W}]_{33} = \delta\theta_1\delta\theta_3w_{13} - w_{33}(\delta\theta_1^2 + \delta\theta_2^2) + \delta\theta_2\delta\theta_3w_{23} \quad (\text{E.14})$$

Therefore, the sum of Eq. E.12, Eq. E.13, and Eq. E.14 is the trace of the 3×3 matrix $[-\boldsymbol{\delta\theta} \times]^2 \mathbf{W}$. Computing the trace and grouping like terms,

$$\begin{aligned} \text{tr} [[-\boldsymbol{\delta\theta} \times]^2 \mathbf{W}] &= -(w_{22} + w_{33})\delta\theta_1^2 - (w_{11} + w_{33})\delta\theta_2^2 \\ &\quad - (w_{11} + w_{22})\delta\theta_3^2 + (w_{21} + w_{12})\delta\theta_1\delta\theta_2 \\ &\quad + (w_{31} + w_{13})\delta\theta_1\delta\theta_3 + (w_{23} + w_{32})\delta\theta_2\delta\theta_3 \end{aligned} \quad (\text{E.15})$$

Therefore, the second derivative of the scalar $\text{tr} [[-\delta\boldsymbol{\theta}\times]^2 \mathbf{W}]$ with respect to $\delta\boldsymbol{\theta}$ may be computed as

$$\begin{aligned} & \frac{\partial^2}{\partial\delta\boldsymbol{\theta}\partial\delta\boldsymbol{\theta}} (\text{tr} [[-\delta\boldsymbol{\theta}\times]^2 \mathbf{W}]) \\ &= \begin{bmatrix} -2(w_{22} + w_{33}) & (w_{21} + w_{12}) & (w_{31} + w_{13}) \\ (w_{21} + w_{12}) & -2(w_{11} + w_{33}) & (w_{23} + w_{32}) \\ (w_{31} + w_{13}) & (w_{23} + w_{32}) & -2(w_{11} + w_{22}) \end{bmatrix} \end{aligned} \quad (\text{E.16})$$

which is equivalent to

$$\frac{\partial^2}{\partial\delta\boldsymbol{\theta}\partial\delta\boldsymbol{\theta}} (\text{tr} [[-\delta\boldsymbol{\theta}\times]^2 \mathbf{W}]) = -2\text{tr} [\mathbf{W}] \mathbf{I}_{3\times 3} + \mathbf{W} + \mathbf{W}^T \quad (\text{E.17})$$

Recalling that $\mathbf{W} \approx \mathbf{W}^T$ produces the following relation

$$\frac{\partial^2}{\partial\delta\boldsymbol{\theta}\partial\delta\boldsymbol{\theta}} (\text{tr} [[-\delta\boldsymbol{\theta}\times]^2 \mathbf{W}]) = -2\text{tr} [\mathbf{W}] \mathbf{I}_{3\times 3} + 2\mathbf{W} \quad (\text{E.18})$$

Now, substituting Eq. E.18 into Eq. E.9 yields

$$\mathcal{F}_{\theta\theta} = -\frac{1}{2}E [-2\text{tr} [\mathbf{W}] \mathbf{I}_{3\times 3} + 2\mathbf{W}] \quad (\text{E.19})$$

Finally, substituting for \mathbf{W} from Eq. E.7, yields

$$\mathcal{F}_{\theta\theta} = \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \mathbf{I}_{3\times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (\text{E.20})$$

This completes the detailed derivation of Eq. 5.54 in Chapter 5.

E.2 Derivation of \mathbf{B} as function of $\mathcal{F}_{\theta\theta}$

The objective of this section is to prove the following result

$$\mathbf{B} = \left[\frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3\times 3} - \mathcal{F}_{\theta\theta} \right] \mathbf{T}_{\hat{B}}^I \quad (\text{E.21})$$

This may be shown by solving Eq. E.20 (or Eq. 5.54) for \mathbf{B} . Therefore, recall from Eq. E.20 that

$$\mathcal{F}_{\theta\theta} = \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \mathbf{I}_{3 \times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (\text{E.22})$$

Taking the trace of both sides shows that

$$\text{tr} [\mathcal{F}_{\theta\theta}] = 3\text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] - \text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] \quad (\text{E.23})$$

And, therefore,

$$\text{tr} [\mathbf{T}_{\hat{B}}^I \mathbf{B}^T] = \frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \quad (\text{E.24})$$

Substituting this result back into Eq. E.22,

$$\mathcal{F}_{\theta\theta} = \frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3 \times 3} - \mathbf{T}_{\hat{B}}^I \mathbf{B}^T \quad (\text{E.25})$$

Rearranging shows that

$$\mathbf{T}_{\hat{B}}^I \mathbf{B}^T = \frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3 \times 3} - \mathcal{F}_{\theta\theta} \quad (\text{E.26})$$

Recalling that $\mathcal{F}_{\theta\theta}$ is symmetric, the transpose of Eq. E.26 is given by

$$\mathbf{B} \mathbf{T}_{\hat{B}}^I = \frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3 \times 3} - \mathcal{F}_{\theta\theta} \quad (\text{E.27})$$

Now, multiplying through on the right by $\mathbf{T}_{\hat{B}}^I$ yields

$$\mathbf{B} = \left[\frac{1}{2} \text{tr} [\mathcal{F}_{\theta\theta}] \mathbf{I}_{3 \times 3} - \mathcal{F}_{\theta\theta} \right] \mathbf{T}_{\hat{B}}^I \quad (\text{E.28})$$

This completes the detailed derivation of Eq. 5.55 in Chapter 5.

Bibliography

- [1] Thornton, C. and Border, J., *Radiometric Tracking Techniques for Deep Space Navigation*, Wiley Interscience, Hoboken, NJ, 2003.
- [2] Sniffin, R., “DSN Telecommunications Link Design Handbook,” DSN No. 810-005, Rev. E, Interplanetary Network Directorate, Jet Propulsion Laboratory, Jan 2001.
- [3] Martin-Mur, T., Bhaskaran, S., Cesarone, R., and McElrath, T., “The Next 25 Years of Deep Space Navigation,” *AAS Guidance and Control Conference*, Breckenridge, CO, 1-6 Feb 2008.
- [4] Abraham, D., “Future Mission Trends and Their Implications for the Deep Space Network,” *AIAA Space 2006*, San Diego, CA, 19-21 Sept 2006.
- [5] MacNeal, B., Abraham, D., and Cesarone, R., “DSN Antenna Array Architectures Based on Future NASA Mission Needs,” *IEEE Aerospace Conference*, Big Sky, MT, 3-10 March 2007.
- [6] “NASA Breaks Ground on New Deep Space Network Antennas,” Jet Propulsion Laboratory, <http://www.jpl.nasa.gov/news/news.cfm?release=2010-065>, 25 Feb 2010, [cited 25 February 2010].

- [7] Garner, R., “Goddard Unveils New Antenna Network in White Sands,” *GoddardView*, Vol. 3, No. 16, 2003, pp. 2–3.
- [8] Goswami, J. and Annadurai, M., “Chandrayaan-1 Mission to the Moon,” *Acta Astronautica*, Vol. 63, No. 11-12, 2008, pp. 1215–1220.
- [9] Orloff, R., “Apollo by the Numbers: A Statistical Reference,” NASA-SP-2000-4029, National Aeronautics and Space Administration, 2000.
- [10] “Report of the 90-Day Study on Human Exploration of the Moon and Mars,” NASA-TM-102999, National Aeronautics and Space Administration, Nov 1989.
- [11] “Report of the Advisory Committee On the Future of the U.S. Space Program,” NASA History Division, Dec 1990.
- [12] “America at the Threshold: America’s Space Exploration Initiative,” National Aeronautics and Space Administration, May 1991.
- [13] Hogan, T., “Mars Wars: The Rise and Fall of the Space Exploration Initiative,” NASA SP-2007-4410, NASA History Division, May 2007.
- [14] Drake, B., “Exploration Blueprint Data Book,” NASA/TM-2007-214763, National Aeronautics and Space Administration, July 2007.
- [15] Hoffman, S. and Kaplan, D., “Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team,” NASA-TM-6107, National Aeronautics and Space Administration, July 1997.

- [16] “Report, Volume 1,” Columbia Accident Investigation Board, Aug. 2003.
- [17] “Vision for Space Exploration,” NASA NP-2004-01-334-HQ, National Aeronautics and Space Administration, Feb 2004.
- [18] “Seeking a Human Spaceflight Program Worthy of a Great Nation,” Review of U.S. Human Spaceflight Plans Committee, Oct 2009.
- [19] Liebe, C., “Accuracy Performance of Star Trackers - A Tutorial,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 2, April 2002, pp. 587–599.
- [20] Sheikh, S., Pines, D., Ray, P., Wood, K., Lovellette, M., and Wolff, M., “Spacecraft Navigation Using X-Ray Pulsars,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, 2006, pp. 49–63.
- [21] Sobel, D., *Longitude*, Walker Publishing Company, New York, NY, 1995.
- [22] Karl, J., *Celestial Navigation in the GPS Age*, Paradise Cay Publications, Arcata, CA, 2009.
- [23] Lillestrand, R. and Carroll, J., “Horizon-Based Satellite Navigation Systems,” *IEEE Transactions on Aerospace and Navigational Electronics*, Vol. 10, No. 3, 1963, pp. 247–270.
- [24] Battin, R., *Astronautical Guidance*, McGraw-Hill, New York, NY, 1964.
- [25] Silva, R., Jorris, T., and Vallerie, E., “Experiment D009 - Simple Navigation,” *The Gemini Program: Physical Sciences Experiments Summary*,

NASA TM X-58075, National Aeronautics and Space Administration, Sept 1971, pp. 37–49.

- [26] Smith, D., “Experiment T002 - Manual Navigation Sightings,” *The Gemini Program: Physical Sciences Experiments Summary*, NASA TM X-58075, National Aeronautics and Space Administration, Sept 1971, pp. 61–69.
- [27] Silva, R., Jorris, T., and Vallerie, E., “Experiment D005 - Star-Occultation Navigation,” *The Gemini Program: Physical Sciences Experiments Summary*, NASA TM X-58075, National Aeronautics and Space Administration, Sept 1971, pp. 51–59.
- [28] Hoag, D., “The History of Apollo On-board Guidance, Navigation, and Control,” *The Eagle has Returned*, edited by E. Steinhoff, Vol. 43, Univelt, 1976, pp. 270–300.
- [29] Draper, C., Wrigley, W., Hoag, G., Battin, R., Miller, J., Koso, D., Hopkins, A., and Vander Velde, W., “Space Navigation Guidance and Control, Vol. 1 of 2,” NASA-CR-75543, June 1965.
- [30] Draper, C., Wrigley, W., Hoag, G., Battin, R., Miller, J., Koso, D., Hopkins, A., and Vander Velde, W., “Space Navigation Guidance and Control, Vol. 2 of 2,” NASA-CR-75798, June 1965.
- [31] Kachmar, P. and Wood, L., “Space Navigation Applications,” *Navigation: Journal of the Institute of Navigation*, Vol. 42, No. 1, 1995,

pp. 187–234.

- [32] Mathematical Physics Branch, “Apollo 15 Navigation Results,” MSC Internal Note No. 71-FM-372, Oct. 1971.
- [33] “Guidance System Operations Plan for Manned CM Earth Orbital and Lunar Missions Using Program Colossus I (Rev. 237) and Program Colossus IA (Rev. 249), Section 5 Guidance Equations (Rev. 4),” GSOP #R-577, Dec. 1968.
- [34] Battin, R., “Some Funny Things Happened on the Way to the Moon,” *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 1–7.
- [35] Getchius, J., Crain, T., and D’Souza, C., “Optical Navigation for the Orion Vehicle,” *18th AAS/AIAA Space Flight Mechanics Meeting*, Galveston, TX, 27-31 Jan 2008.
- [36] Walsh, R. and Ferguson, J., “U.S. Air Force Manual Space Navigation Experiment on Skylab (DoD/NASA Skylab Experiment T-002),” *AIAA/AAS Astrodynamics Conference*, San Diego, CA, 18-20 Aug 1976.
- [37] Jorris, T. and Barth, A., “The USAF Manned Space Navigation Experiment on Apollo and its Implications on Advanced Manned Spacecraft,” *ION 1968 National Space Meeting Proceedings*, Cocoa Beach, FL, 19-21 Feb 1968.

- [38] Randle, R. and Powers, S., “The Background for Skylab Experiment T-002, Manual Navigation Sightings,” NASA-TM-X-62420, March 1975.
- [39] Kachmar, P., “Apollo and Space Shuttle On-Board Navigation Systems: Applications of Kalman Filtering Techniques,” *ION 58th Annual Meeting / CIGTF 21st Guidance Test Symposium*, Albuquerque, NM, 24-26 June 2002.
- [40] Duxbury, T. and Breckenridge, W., “Mariner Mars 1969 Optical Approach Navigation,” *AIAA 8th Aerospace Sciences Meeting*, New York, NY, 19-21 Jan 1970.
- [41] Action, C., “Processing Onboard Optical Data for Planetary Approach Navigation,” *Journal of Spacecraft and Rockets*, Vol. 9, No. 10, 1972, pp. 746–750.
- [42] Duxbury, T., Born, G., and Jerath, N., “Viewing Phobos and Deimos for Navigation on Mariner 9,” *Journal of Spacecraft and Rockets*, Vol. 11, No. 4, 1974, pp. 215–222.
- [43] Jerath, N. and Ohtakay, H., “Mariner IX Optical Navigation Using Mars Lit Limb,” *Journal of Spacecraft and Rockets*, Vol. 11, No. 7, 1974, pp. 505–511.
- [44] Owen, W., Duxbury, T., Action, C., Synnott, S., Riedel, J., and Bhaskaran, S., “A Brief History of Optical Navigation at JPL,” *AAS Guidance and Control Conference*, Breckenridge, CO, 1-6 Feb 2008.

- [45] Kohlhasse, C. and Penzo, P., “Voyager Mission Description,” *Space Science Reviews*, Vol. 21, No. 2, 1977, pp. 77–101.
- [46] Rudd, R., Hall, J., and Spradlin, G., “The Voyager Interstellar Mission,” *Acta Astronautica*, Vol. 40, No. 2-8, 1997, pp. 383–396.
- [47] Matson, D., Spilker, L., and Lebreton, J., “The Cassini/Huygens Mission to the Saturnian System,” *Space Science Reviews*, Vol. 104, No. 1-4, 2002, pp. 1–58.
- [48] Mitchell, R., “The Cassini Mission exploring Saturn,” *Acta Astronautica*, Vol. 63, 2008, pp. 61–67.
- [49] Campbell, J., Synnott, S., and Bierman, G., “Voyager Orbit Determination at Jupiter,” *IEEE Transactions on Automatic Control*, Vol. 28, No. 3, 1981, pp. 256–268.
- [50] Riedel, J., Owen, W., and Stuve, J., “Optical Navigation During the Voyager Neptune Encounter,” *AIAA/AAS Astrodynamics Conference*, Portland, OR, 20-22 Aug 1990.
- [51] Synnott, S., Donegan, A., Riedel, J., and Stuve, J., “Interplanetary Optical Navigation: Voyager Uranus Encounter,” *AIAA Astrodynamics Conference*, Williamsburg, VA, 18-20 Aug 1986.
- [52] Gillam, S., Owen, W., Vaughan, A., Wang, T.-C., Costello, J., Jacobson, R., Bluhm, D., Pojman, J., and Ionasescu, R., “Optical Navigation

for the Cassini/Huygens Mission,” *AAS/AIAA Astrodynamics Specialist Conference*, Mackinac Island, MI, 19-23 Aug 2007.

- [53] Johnson, M., “The Galileo High Gain Antenna Deployment Failure,” *Proceedings of the 28th Aerospace Mechanisms Symposium*, NASA Lewis Research Center, 18-20 May 1994, pp. 359–377.
- [54] Hoffman, A., Green, N., and Garrett, H., “Assessment of In-Flight Anomalies of Long Life Outer Planet Missions,” *Proceedings of the 5th International Symposium on Environmental Testing for Space Programmes*, Noordwijk, The Netherlands, 15-17 June 2004.
- [55] Vaughan, R., Riedel, J., Davis, R., Owen, W., and Synnott, S., “Optical Navigation for the Galileo Gaspra Encounter,” *AIAA/AAS Astrodynamics Conference*, Hilton Head Island, SC, 10-12 Aug 1992.
- [56] Bhaskaran, S., Riedel, J., and Synnott, S., “Autonomous Optical Navigation for Interplanetary Missions,” *SPIE*, Vol. 2810, 1996, pp. 32–43.
- [57] “PIA00273: Optical Navigation Image of Ganymede,” Photojournal, Jet Propulsion Laboratory, California Institute of Technology, <http://photojournal.jpl.nasa.gov/catalog/PIA00273>, [cited 29 November 2009].
- [58] Santo, A., Lee, S., and Gold, R., “NEAR Spacecraft and Instrumentation,” *Journal of the Astronautical Sciences*, Vol. 43, No. 4, 1995, pp. 373–397.

- [59] Williams, B., Antreasian, P., Bordi, J., Carranza, E., Chesley, S., Helfrich, C., Miller, J., Owen, W., and Wang, T., “Navigation Results for NEAR Shoemaker: The First Spacecraft to Orbit an Asteroid,” *AAS/AIAA Astrodynamics Conference*, Quebec City, Quebec, 30 July - 2 Aug 2001.
- [60] Owen, W., Wang, T., Harch, A., Bell, M., and Peterson, C., “NEAR Optical Navigation at Eros,” *AAS/AIAA Astrodynamics Conference*, Quebec City, Quebec, 30 July - 2 Aug 2001.
- [61] Rayman, M., Varghese, P., Lehman, D., and Livesay, L., “Results from the Deep Space 1 Technology Validation Mission,” *Acta Astronautica*, Vol. 47, No. 2-9, 2000, pp. 475–487.
- [62] Riedel, J., Bhaskaran, S., Synnott, S., Desai, S., Bollman, W., Dumont, P., Halsell, C., Han, D., Kennedy, B., Null, G., Owen, W., Werner, R., and Williams, B., “Navigation for the New Millennium: Autonomous Navigation for Deep Space 1,” *Proceedings of the 12th International Symposium on Space Flight Dynamics*, Darmstadt, Germany, 2-6 June 1997, pp. 303–320.
- [63] Bhaskaran, S., Riedel, J., and Synnott, S., “Autonomous Nucleus Tracking for Comet/Asteroid Encounters: The STARDUST Example,” *Proceedings of the IEEE Aerospace Conference*, 21-28 March 1998, pp. 353–365.

- [64] Riedel, J., Bhaskaran, S., Desai, S., Han, D., Kennedy, B., McElrath, T., Null, G., Ryne, M., Synnott, S., Wang, T., and Werner, R., “Using Autonomous Navigation for Interplanetary Missions: The Validation of Deep Space 1 AutoNav,” *IAA International Conference on Low-Cost Planetary Missions*, Laurel, MD, 2-5 May 2000.
- [65] Mastrodemos, N., Kubitschek, D., and Synnott, S., “Autonomous Navigation for the Deep Impact Mission Encounter with Comet Tempel 1,” *Space Science Reviews*, Vol. 117, No. 1-2, 2005, pp. 95–121.
- [66] Williams, K., Taylor, A., Stanbridge, D., Wolff, P., Page, B., Williams, B., and McAdams, J., “Navigation for the MESSENGER Missions First Mercury Encounter,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, No. AIAA Paper 2008-6761, Honolulu, HI, 18-21 Aug 2008.
- [67] Hash, C., “MESSENGER MDIS CALIBRATED (CDR) DATA E/V/H V1.0,” NASA Planetary Data System, 2008.
- [68] Antreasian, P., Baird, D., Border, J., Burkhart, P., Graat, E., Jah, M., Mase, R., McElrath, T., and Portock, B., “2001 Mars Odyssey Orbit Determination During Interplanetary Cruise,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Monterey, CA, 5-8 Aug 2001.
- [69] D’Amario, L., “Mars Exploration Rovers Navigation Results,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Providence, RI, 16-19 Aug 2004.

- [70] Portock, B., Kruizinga, G., Bonfiglio, E., Raofi, B., and Ryne, M., “Navigation Challenges of the Mars Phoenix Lander Mission,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, HI, 18-20 Aug 2008.
- [71] “PIA02699: Optical Navigation Demonstration Near Mars,” Photojournal, Jet Propulsion Laboratory, California Institute of Technology, <http://photojournal.jpl.nasa.gov/catalog/PIA02699>, [cited 29 November 2009].
- [72] Hall, R., “Lunar Impact: A History of Project Ranger,” NASA-SP-4210, National Aeronautics and Space Administration, 1977.
- [73] “Surveyor Program Results,” NASA-SP-184, National Aeronautics and Space Administration, 1969.
- [74] Lozier, D., Galal, K., Folta, D., and Beckman, M., “Lunar Prospector Mission Design and Trajectory Support,” AAS 98-323, 1998.
- [75] Regeon, P., Lynn, P., and Johnson, M., “The Clementine Lunar Orbiter,” *20th International Symposium on Space Technology and Science*, Gifu, Japan, 19-25 May 1996.
- [76] Sorensen, T., Oswald, D., Shook, R., and Van Gaasbeck, J., “Spacecraft Autonomous Operations Experiment Performed During the Clementine Lunar Mission,” *Journal of Spacecraft and Rockets*, Vol. 32, No. 6, 1995, pp. 1049–1053.

- [77] Chin, G., Brylow, S., Foote, M., Garvin, J., Kasper, J., Keller, J., Litvak, M., Mitrofanov, I., Paige, D., Raney, K., Robinson, M., Sanin, A., Smith, D., Spence, H., Spudis, P., Stern, S., and Zuber, M., “Lunar Reconnaissance Orbiter Overview: The Instrument Suite and Mission,” *Space Science Reviews*, Vol. 129, 2007, pp. 391–419.
- [78] Wertz, J., editor, *Spacecraft Attitude Determination and Control*, D. Reidel Publishing Company, Dordrecht, Holland, 1985.
- [79] Shuster, M., “A Survey of Attitude Representations,” *The Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517.
- [80] Kuipers, J., *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*, Princeton University Press, Princeton, NJ, 1999.
- [81] Crassidis, J. and Junkins, J., *Optimal Estimation of Dynamical Systems*, CRC Press LLC, Boca Raton, FL, 2004.
- [82] Van Steenwyk, B., “Using the Matrix Exponential Solution to the Direction Cosine Differential Equation,” *IEEE Position Location and Navigation Symposium*, Atlanta, GA, 22-26 April 1996.
- [83] Zanetti, R., *Advanced Navigation Algorithms for Precision Landing*, Ph.D. thesis, The University of Texas at Austin, December 2007.
- [84] Penna, M. and Patterson, R., *Prospective Geometry and its Applications to Computer Graphics*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

- [85] Salomon, D., *Transformations and Projections in Computer Graphics*, Springer, London, 2006.
- [86] Ma, Y., Soatto, S., Košecká, J., and Sastry, S., *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer, New York, NY, 2004.
- [87] Forsyth, D. and Ponce, J., *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, 2003.
- [88] Schroeder, D., *Astronomical Optics, Second Edition*, Academic Press, San Diego, CA, 2000.
- [89] Stauder, J., Lowman, A., Thiessen, D., Day, D., and Miles, D., “Off-axis Scatter Measurements of the Mars Reconnaissance Orbiter (MRO) Optical Navigation Camera (ONC),” *SPIE*, Vol. 5874, 2005, pp. 208–219.
- [90] Lowman, A. and Stauder, J., “Stray light lessons learned from the Mars Reconnaissance Orbiters Optical Navigation Camera,” *SPIE*, Vol. 5526, 2004, pp. 240–248.
- [91] McConnell, J., *Computer Graphics: Theory into Practice*, Jones and Bartlett Publishers, Boston, MA, 2006.
- [92] Janesick, J., *Scientific Charge-Coupled Devices*, SPIE Press, Bellingham, WA, 2001.

- [93] Hash, C., “MESSENGER MDIS EXPERIMENT (EDR) DATA E/V/H V1.0,” NASA Planetary Data System, 2008.
- [94] Naval Research Laboratory, “CLEM1-L/E/Y-A/B/U/H/L/N-2-EDR-V1.0,” NASA Planetary Data System, 1995.
- [95] Holst, G. and Lomheim, T., *CMOS/CCD Sensors and Camera Systems*, JCD Publishing and SPIE Press, Winter Park, FL, 2007.
- [96] Shuster, M., “Stellar Aberration and Parallax: A Tutorial,” *The Journal of the Astronautical Sciences*, Vol. 51, No. 4, 2003, pp. 447–494.
- [97] Christian, J., Wells, G., Lafleur, J., Verges, A., and Braun, R., “Extension of Traditional Entry, Descent, and Landing Technologies for Human Mars Exploration,” *Journal of Spacecraft and Rockets*, Vol. 45, No. 1, 2008, pp. 130–141.
- [98] Kordas, J., Lewis, I., Wilson, B., Nielsen, D., Park, H., Priest, R., Hills, R., Shannon, M., Ledebuhr, A., and Pleasance, L., “Star Tracker Stellar Compass for the Clementine Mission,” *SPIE*, Vol. 2466, 1995, pp. 70–83.
- [99] “Laser Black Properties and Tech Data,” Epner Technology, http://www.epner.com/laser_black.ssi, [cited 05 February 2010].
- [100] Leinert, C. and Klüppelberg, D., “Stray Light Suppression in Optical Space Experiments,” *Applied Optics*, Vol. 13, No. 3, March 1974, pp. 556–564.

- [101] Lewis, I., Ledebuhr, A., Axelrod, T., and Ruddell, S. A., “Stray Light Reduction in a WFOV Star Tracker Lens,” *SPIE*, Vol. 1530, 1991, pp. 306–324.
- [102] Bely, P., *The Design and Construction of Large Optical Telescopes*, Springer, 2003.
- [103] Mahajan, V., *Optical Imaging and Aberrations, Part I: Ray Geometrical Optics*, SPIE Optical Engineering Press, 1998.
- [104] Devernay, F. and Faugeras, O., “Automatic Calibration and Removal of Distortion from Scenes of Structured Environments,” *SPIE*, Vol. 2567, 1995, pp. 62–72.
- [105] Pentland, A., “A New Sense for Depth of Field,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, 1987, pp. 523–531.
- [106] Chaudhuri, S. and Rajagopalan, A., *Depth From Defocus: A Real Aperture Imaging Approach*, Springer, 1999.
- [107] Martinez, P. and Klotz, A., *A Practical Guide to CCD Astronomy*, Cambridge University Press, Cambridge, UK, 1998.
- [108] Pain, B., Cunningham, T., Hancock, B., Yang, G., Seshadri, S., and Oritz, M., “Reset Noise Suppression in Two-Dimensional CMOS Photodiode Pixels through Column-Based Feedback-Reset,” *2002 International Electronic Devices Meeting*, San Francisco, CA, 8-11 Dec 2002.

- [109] Tsai, T., “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses,” *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, March-April 1987, pp. 323–344.
- [110] Zhuang, H. and Roth, Z., *Camera-Aided Robot Calibration*, CRC Press, Boca Raton, FL, 1996.
- [111] Zhang, Z., “Flexible Camera Calibration by Viewing a Plane from Unknown Orientations,” *IEEE International Conference on Computer Vision*, Kerkyra, Greece, 20-27 Sept 1999.
- [112] Hapke, B., “Bidirectional Reflectance Spectroscopy 1. Theory,” *Journal of Geophysical Research*, Vol. 86, No. B4, 1981, pp. 3039–3054.
- [113] Hapke, B. and Wells, E., “Bidirectional Reflectance Spectroscopy 2. Applications,” *Journal of Geophysical Research*, Vol. 86, No. B4, 1981, pp. 3055–3060.
- [114] Fairbairn, M., “Planetary Photometry: The Lommel-Seeliger Law,” *Journal of the Royal Astronomical Society of Canada*, Vol. 99, No. 3, June 2005, pp. 92–93.
- [115] Pisacane, V., *The Space Environment and its Effect on Space Systems*, AIAA Educational Series, Reston, VA, 2008.
- [116] Hapke, B., “A Theoretical Photometric Function for the Lunar Surface,” *Journal of Geophysical Research*, Vol. 68, No. 15, 1963, pp. 4571–4586.

- [117] Hapke, B., “An Improved Theoretical Lunar Photometric Function,” *The Astronomical Journal*, Vol. 71, No. 5, 1966, pp. 333–339.
- [118] Dereniak, E. and Crowe, D., *Optical Radiation Detectors*, John Wiley & Sons, New York, NY, 1984.
- [119] Bedding, T., Zijlstra, A., von der Lühe, O., Robertson, J., Marson, R., Barton, J., and Carter, B., “The Angular Diameter of R Doradus: A Nearby Mira-like Star,” *Monthly Notices of the Royal Astronomical Society*, Vol. 286, No. 4, 1997, pp. 957–962.
- [120] Shuster, M., “Maximum Likelihood Estimation of Spacecraft Attitude,” *The Journal of the Astronautical Sciences*, Vol. 37, No. 1, Jan-March 1989, pp. 79–88.
- [121] Shuster, M., “Constraint in Attitude Estimation Part 1: Constrained Estimation,” *The Journal of the Astronautical Sciences*, Vol. 51, No. 1, 2003, pp. 51–74.
- [122] Cheng, Y., Crassidis, J., and Markley, F., “Attitude Estimation for Large Field-of-View Sensors,” *Journal of the Astronautical Sciences*, Vol. 54, No. 2-3, 2003, pp. 433–448.
- [123] Shuster, M., “Erratum: Kalman Filtering of Spacecraft Attitude and the QUEST Model,” *The Journal of the Astronautical Sciences*, Vol. 51, No. 3, 2003, pp. 359.

- [124] “STAR1000 1M Pixel Radiation Hard CMOS Image Sensor,” Tech. Rep. 38-05714 Rev. B, Cypress Semiconductor Corporation, Jan 2007.
- [125] Hawkins, S., Boldt, J., Darlington, E., Espiritu, R., Gold, R., Gotwols, B., Grey, M., Hash, C., Hayes, J., Jaskulek, S., Kardian, C., Keller, M., Malaret, E., Murchie, S., Murphy, P., Peacock, K., Prockter, L., Reiter, R., Robinson, M., Schaefer, E., Shelton, R., Sterner, R., Taylor, H., Watters, T., and Williams, B., “The Mercury Dual Imaging System on the MESSENGER Spacecraft,” *Space Science Reviews*, Vol. 131, No. 1-4, 2007, pp. 247–338.
- [126] Sidi, M., *Spacecraft Dynamics & Control, A Practical Engineering Approach*, Cambridge University Press, 2001.
- [127] Johnson, A., Ansar, A., Matthies, L., Trawny, N., Mourikis, A., and Roumeliotis, S., “A General Approach to Terrain Relative Navigation for Planetary Landing,” *Infotech@Aerospace 2007 Conference and Exhibit*, Rohnert Park, CA, 7-10 May 2007.
- [128] Adams, D., Criss, T., and Shankar, U., “Passive Optical Terrain Relative Navigation Using APLNav,” *2008 IEEE Aerospace Conference*, 1-8 March 2008.
- [129] Osenar, M., Clark, F., and D’Souza, C., “Performance of an Automated Feature Tracking Lunar Navigation System,” *18th AAS/AIAA Space Flight Mechanics Meeting*, Galveston, TX, 27-31 Jan 2008.

- [130] Jones, B., “Surface Feature Navigation in Low Lunar Orbit,” *18th AAS/AIAA Space Flight Mechanics Meeting*, Galveston, TX, 27-31 Jan 2008.
- [131] Hanak, F., *Lost in Low Lunar Orbit Crater Pattern Detection and Identification*, Ph.D. thesis, The University of Texas at Austin, May 2009.
- [132] Psiaki, M. and Hinks, J., “Autonomous Lunar Orbit Determination using Star Occultation Measurements,” *AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, 20-23 Aug 2007.
- [133] Landgraf, M., Thiele, G., Koschny, D., and Udrea, B., “Optical Navigation for Lunar Exploration Missions,” *57th International Astronautical Congress*, Valencia, Spain, 2-6 Oct 2006.
- [134] Brannan, D., Esplen, M., and Gray, J., *Geometry*, Cambridge University Press, Cambridge, UK, 1999.
- [135] Zhang, Z., “Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting,” *Image and Vision Computing*, Vol. 15, No. 1, 1997, pp. 59–76.
- [136] Fitzgibbon, A., Pilu, M., and Fisher, B., “Direct Least Squares Fitting of Ellipses,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 5, 1999, pp. 476–480.
- [137] Halíř, R. and Flusser, J., “Numerically Stable Direct Least Squares Fitting of Ellipses,” *6th International Conference in Central Europe on*

Computer Graphics and Visualization (WSCG '98), Bory, Czech Republic, 9-13 Feb 1998.

- [138] Maini, E., “Robust Ellipse-Specific Fitting for Real-Time Machine Vision,” *Lecture Notes in Computer Science*, Vol. 3704, 2005, pp. 318–327.
- [139] Fischler, M. and Bolles, R., “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395.
- [140] Torr, P. and Zisserman, A., “Robust Computation and Parametrization of Multiple View Relations,” *IEEE 6th International Conference on Computer Vision*, Bombay, India, 4-7 Jan 1998.
- [141] Torr, P. and Zisserman, A., “MLESAAC: A New Robust Estimator with Application to Estimating Image Geometry,” *Computer Vision and Image Understanding*, Vol. 78, No. 1, 2000, pp. 138–156.
- [142] Huber, P., *Robust Statistics*, John Wiley and Sons, New York, NY, 1981.
- [143] Wilcox, R., *Introduction to Robust Estimation and Hypothesis Testing*, Academic Press, San Diego, CA, 1997.
- [144] Mortari, D., “Search-Less Algorithm for Star Pattern Recognition,” *The Journal of the Astronautical Sciences*, Vol. 45, No. 2, April-June 1997, pp. 179–194.

- [145] Mortari, D., Samaan, M., Bruccoleri, C., and Junkins, J., “The Pyramid Star Identification Technique,” *Navigation*, Vol. 51, No. 3, 2004, pp. 171–183.
- [146] Koenderink, J. and Pont, S., “Irradiation Direction from Texture,” *Journal of the Optical Society of America, A: Optics, Image Science, and Vision*, Vol. 20, No. 10, 2003, pp. 1875–1882.
- [147] Varma, M. and Zisserman, A., “Estimating Illumination Direction from Textured Images,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.
- [148] Roatsch, T., “Mars Express HRSC Radiometric RDR V1.0, MEX-M-HRSC-3-RDR-V2.0,” European Space Agency, 2004.
- [149] Enright, J., “Moon-Tracking Modes for Star Trackers,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, Jan-Feb 2010, pp. 171–185.
- [150] Tapley, B., Schutz, B., and Born, G., *Statistical Orbit Determination*, Elsevier Academic Press, Burlington, MA, 2004.
- [151] Brown, R. and Hwang, P., *Introduction to Random Signals and Applied Kalman Filtering, Third Edition*, John Wiley & Sons, New York, NY, 1997.
- [152] Gelb, A., *Applied Optimal Estimation*, The MIT Press, Cambridge, MA, 1974.

- [153] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, AIAA Educational Series, Reston, VA, 1999.
- [154] Wahba, G., “A Least Squares Estimate of Satellite Attitude,” *SIAM Review*, Vol. 7, No. 3, July 1965, pp. 409.
- [155] Keat, J., “Analysis of Least-Squares Attitude Determination Routine DOAOP,” Tech. Rep. CSC/TM-77/6034, Computer Sciences Corp, Feb 1977.
- [156] Markley, F. and Mortari, D., “Quaternion Attitude Estimation Using Vector Observations,” *The Journal of the Astronautical Sciences*, Vol. 48, No. 2/3, April-September 2000, pp. 359–380.
- [157] Shuster, M. and Oh, S., “Three-Axis Attitude Determination from Vector Observations,” *Journal of Guidance and Control*, Vol. 4, No. 1, January-February 1981, pp. 70–77.
- [158] Mortari, D., “ESOQ: A Closed-Form Solution to the Wahba Problem,” *The Journal of the Astronautical Sciences*, Vol. 45, No. 2, April-June 1997, pp. 195–204.
- [159] Mortari, D., “ESOQ-2 Single-Point Algorithm for Fast Optimal Spacecraft Attitude Determination,” *Advances in the Astronautical Sciences*, Vol. 95, No. 2, 1997, pp. 817–826.

- [160] Markley, F., "Attitude Determination Using Vector Observations and the Singular Value Decomposition," *The Journal of the Astronautical Sciences*, Vol. 36, No. 3, 1988, pp. 245–258.
- [161] Shuster, M., "A Simple Kalman Filter and Smoother for Spacecraft Attitude," *Journal of the Astronautical Sciences*, Vol. 37, No. 1, January-March 1989, pp. 89–106.
- [162] Shuster, M., "New Quests for Better Attitudes," *Flight Mechanics/Estimation Theory Symposium*, NASA Goddard Space Flight Center, 21-23 May 1991.
- [163] Bar-Itzhack, I., "REQUEST: A Recursive QUEST Algorithm for Sequential Attitude Determination," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, Sept-Oct 1996, pp. 1034–1038.
- [164] Shuster, M., "Filter QUEST or REQUEST," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, March-April 2009, pp. 643–645.
- [165] Choukroun, D., Bar-Itzhack, I., and Oshman, Y., "Optimal-REQUEST Algorithm for Attitude Determination," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, May-June 2004, pp. 418–425.
- [166] Psiaki, M., "Attitude-Determination Filtering via Extended Quaternion Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 2, March-April 2000, pp. 206–214.

- [167] Markley, F., “Attitude Determination and Parameter Estimation Using Vector Observations: Theory,” *The Journal of the Astronautical Sciences*, Vol. 37, No. 1, January-March 1989, pp. 41–58.
- [168] Markley, F., “Attitude Determination and Parameter Estimation Using Vector Observations: Application,” *The Journal of the Astronautical Sciences*, Vol. 39, No. 3, July-September 1991, pp. 367–381.
- [169] Crassidis, J., Markley, F., and Cheng, Y., “Survey of Nonlinear Attitude Estimation Methods,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 12–28.
- [170] Wiesel, W., *Spaceflight Dynamics, Second Edition*, Irwin/McGraw-Hill, Boston, MA, 1997.
- [171] Hull, D., *Optimal Control Theory for Applications*, Springer, New York, NY, 2003.
- [172] Mortari, D., “n-Dimensional Cross Product and its Application to the Matrix Eigenanalysis,” *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, 1997, pp. 509–515.
- [173] Sorenson, H., *Parameter Estimation: Principles and Problems*, Marcel Dekker, Inc, New York, NY, 1980.
- [174] Rencher, A., *Multivariate Statistical Inference and Applications*, John Wiley and Sons, Inc., New York, NY, 1998.

- [175] Morrison, D., *Multivariate Statistical Methods, Fourth Edition*, Brooks/Cole, Belmont, CA, 2005.
- [176] Markley, F., “Attitude Error Representations for Kalman Filtering,” *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, March-April 2003, pp. 311–317.
- [177] Atkinson, K., *An Introduction to Numerical Analysis, Second Edition*, John Wiley & Sons, New York, NY, 1989.
- [178] Muñoz, S., Christian, J., and Lightsey, E., “Development of an End to End Simulation Tool for Autonomous Cislunar Navigation,” *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, 10-13 August 2009.
- [179] Farrenkopf, R., “Analytic Steady-State Accuracy Solutions for Two Common Spacecraft Attitude Estimators,” *Journal of Guidance and Control*, Vol. 1, No. 4, 1978, pp. 282–284.
- [180] Kalman, R., “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.
- [181] Kalman, R. and Bucy, R., “New Results in Linear Filtering and Prediction,” *Journal of Basic Engineering*, Vol. 83, 1961, pp. 95–108.
- [182] Zarchan, P. and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach, Second Edition*, Progress in Astronautics and Aeronau-

- tics, Vol. 208, American Institute of Aeronautics and Astronautics, Reston, VA, 2005.
- [183] Vallado, D., *Fundamentals of Astrodynamics and Applications, 3rd Edition*, Springer, New York, NY, 2007.
- [184] Standish, E., “JPL Planetary and Lunar Ephemerides, DE405/LE405,” Tech. Rep. IOM 312.F-98-048, Jet Propulsion Laboratory, Aug 1998.
- [185] Hash, C., Espiritu, R., Malaret, E., Prockter, L., Murchie, S., Mick, A., and Ward, J., “MESSENGER Mercury Dual Imaging System (MDIS) Experimental Data Record (EDR) Software Interface Specification (SIS),” NASA PDS Imaging Node. http://pds-imaging.jpl.nasa.gov/data/messenger/msgrmds_1001/DOCUMENT/MDISEDRSIS.PDF, Nov 2008.
- [186] Proud, R., Hardy, J., Pape, M., and Rosburg, M., “Onboard Automated GN&C Decision-Making Concepts for NASA’s CEV,” AAS 07-074, Feb. 2007.
- [187] Zanetti, R., “Autonomous Midcourse Navigation for Lunar Return,” *Journal of Spacecraft and Rockets*, Vol. 46, No. 4, July-Aug 2009, pp. 865–873.
- [188] “MIMU Miniature Inertial Measurement Unit,” Tech. Rep. 03-S-1918, Honeywell, Sept 2003.
- [189] “NASA Systems Engineering Handbook, Rev 1,” NASA/SP-2007-6105, National Aeronautics and Space Administration, Dec 2007.

- [190] “NASA Research and Technology Program and Project Management Requirements,” NASA Procedural Requirements (NPR) 7120.8, National Aeronautics and Space Administration, Feb 2008.
- [191] “SMC Systems Engineering Primer and Handbook, Third Edition,” Space and Missile Systems Center, U.S. Air Force, April 2005.
- [192] Regehr, M., Acikmese, A., Ahmed, A., Aung, M., Bailey, R., Bushnell, C., Clark, K., Hicke, A., Lytle, B., MacNeal, P., Rasmussen, R., Shields, J., and Singh, G., “The Formation Control Testbed,” *2004 IEEE Aerospace Conference*, Big Sky, MT, 6-13 March 2004.
- [193] Jung, D. and Tsiotras, P., “A 3-DoF Experimental Test-Bed for Integrated Attitude Dynamics and Control Research,” *AIAA Guidance, Navigation and Control Conference*, Austin, TX, 11-14 Aug 2003.
- [194] “JSC Reduced Gravity Program Photographs,” NASA Johnson Space Center, <http://zerog.jsc.nasa.gov/>, [cited 9 March 2010].
- [195] Christian, J., Turbe, M., Kabo, E., Manno, L., and Johnson, E., “Development of a Variable Inertia Reaction Wheel System for Spacecraft Attitude Control,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, 16-19 Aug 2004.
- [196] Garner, M., *Systems Engineering Processes for a Student-based Design Laboratory*, Master’s thesis, The University of Texas at Austin, December 2009.

- [197] STS-127 Shuttle Mission Imagery , “National Aeronautics and Space Administration,” <http://spaceflight.nasa.gov/gallery/images/shuttle/sts-127/ndxpage63.html>, [cited 05 March 2010].
- [198] Minnaert, M., “The Reciprocity Principle in Lunar Photometry,” *The Astrophysical Journal*, Vol. 93, 1941, pp. 403–410.
- [199] Kanas, N., *Star Maps: History, Artistry, and Cartography*, Praxis Publishing, Chichester, UK, 2009.
- [200] Smith, N., Bae, S., and Schutz, B., “Forty-Nine Biased Star Positions from ICESat Flight Data,” *AAS Space Flight Mechanics Meeting*, San Diego, CA, 14-17 Feb 2010.
- [201] Sande, C., Warren, W., and Tracewell, D., “Recent Enhancements to and Future Plans for the SKY2000 Star Catalog,” *Journal of the American Association of Variable Star Observers*, Vol. 29, No. 4, 2001, pp. 123–128.
- [202] European Space Agency, “The Hipparcos and Tycho Catalogues - Astrometric and Photometric Star Catalogues derived from the ESA Hipparcos Space Astrometry Mission,” Sp-1200, ESA Publications Division, June 1997.
- [203] Perryman, M., “The HIPPARCOS Catalogue Contents,” *ESA Symposium ‘Hipparcos - Venice ’97’*, Venice, Italy, 13-16 May 1997.

- [204] Hog, E., Fabricius, C., Makarov, V., Urban, S., Corbin, T., Wycoff, G., Bastian, U., Schwekendiek, P., and Wicenec, A., “The Tycho-2 Catalogue of the 2.5 Million Brightest Stars,” *Astronomy and Astrophysics*, Vol. 355, 2000, pp. L28–L30.
- [205] Greenberg, M., *Advanced Engineering Mathematics, Second Edition*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1998.
- [206] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B., *Numerical Recipes, The Art of Scientific Computing, Third Edition*, Cambridge University Press, Cambridge, UK, 2007.

Index

- Aberrations, Optical, 85
- Abstract, viii
- Acknowledgments*, v
- Advancement Degree of Difficulty, 295
- Airy Disk, 87
- Albedo, Bond, 103, 105
- Apollo Program, 20–23, 139, 279
- Attitude Profile Matrix, 216

- Background pixels, 151
- Baffle, Light, 77–78
- Bayes Theorem, 227
- Bibliography, 337–365
- Bidirectional Reflectance, 102
- Blooming, 66–68, 196

- Calibration, Camera, 94–96
- Cassini, 27–32, 95
- Charged-Coupled Devices (CCD), 65–68
- Clementine Spacecraft, 48, 78
- Closing an image, 153
- Complementary Metal Oxide Semiconductors (CMOS), 68–70
- Conclusions, 310–313
- Connected Components, 153–155
- Constellation Program, 5
- Contributions, 11–14
- Convolution, 148
- Cramér-Rao Inequality, 222
- Cross-correlation, 148

- Davenport Matrix, 219
- Davenport’s q-Method, 220
- Deep Impact, 44
- Deep Space 1, 40–44
- Deep Space Network (DSN), 2, 39
- Dilation, 152–153
- Distance to an Ellipse
 - Algebraic, 162–163
 - Geometric, 159–162

- Ellipse Fitting, 163–170
- Ephemeris Files, 267
- Erosion, 152–153
- Exposure Time, 194–197
- Extended Kalman Filter(EKF), 212, 242, 263–265
- Extended-QUEST, 211, 251–253

- FASTRAC, 307
- Fill Factor, 70
- Filter QUEST, 210, 246–248
- Fisher Information Matrix, 222–223
- Foreground pixels, 151
- Frontal Image Plane, 63–65, 268

- Galileo, 32–39
- Gemini Program, 17–20

- Homogeneous Coordinates, 53–54
- Hough transform, 164

- Interfaces, 301–303
- International Space Station, 5, 300

- Kalman Filter, 263
- Kernel, image filter, 148–149

Lambertian reflection, 102
 Lommel-Seeliger Law, 102, 196
 Lunar Reconnaissance Orbiter (LRO),
 48
 M-Estimator SAmple Consensus (MSAC), Opening an image, 153
 173–174
 Mahalanobis Distance, 225
 Maritime Navigation, 15–16
 Mars Exploration Rovers (MER), 47
 Mars Odyssey, 46
 Mars Reconnaissance Orbiter (MRO),
 47, 61, 78, 301
 Measurement Sensitivity Matrix, 204
 Measurements, Optical
 Apparent Diameter and Centroid,
 137–139
 Horizon-Reference Star, 139–140
 Surface Feature-Reference Star,
 141–142
 Time of Star Occultation, 143–
 146
 MESSENGER, 44–46, 267–278
 Misalignment
 Camera Mounting, 80–81
 Detector Plane, 81–83
 NEAR Mission, 39–40
 New Horizons, 3
 Noise Floor, 91
 Noise, Detector
 ADC Quantization Noise, 93–
 94
 Amplifier Noise, 91–92
 Dark Current, 89–90
 Fixed Pattern Noise, 93
 Photo-Response Non-Uniformity,
 93
 Reset Noise, 92
 Shot Noise, 91
 Optimal-REQUEST, 211, 249–251
 PARADIGM, 307
 Parallax, 74–76
 Pathfinder, Mars, 43, 46
 Phoenix Mars Lander, 47
 Pinhole Camera Model, 60
 Planck’s Law, 107
 Preliminary Design Review, 298
 Pulsars, Navigation with, 10–11
 Quantum Efficiency, 70
 Quaternion, 52, 317–320
 QUEST, 209, 220, 281, 326–330
 Radial Distortion, 85–86
 RANdOm SAmple Consensus (RANSAC),
 171–172
 Rayleigh Criterion, 87
 REQUEST, 210, 246
 Rodrigues Parameters, 51
 Sequential Optimal Attitude Recur-
 sion (SOAR) Filter, 208, 212–
 213, 226–259, 281
 Sextant
 Apollo, 21
 Gemini, 18
 Skylab Program, 23
 Space Exploration Initiative (SEI),
 4

Space Shuttle Program, 23–24
Star Catalogs, 321
Stardust, 43
Stellar Aberration, 71–73
Stray Light, 76–80
Sun Exclusion Angle, 77, 181

Technology Readiness Level (TRL),
 295–297
Thin Lens, 59
Thresholding, 150–151

Vision for Space Exploration (VSE),
 5
Voyager, 27–32

Wahba Problem, 209, 214–215

Vita

John Allen Christian, III, was born in Groton, Connecticut, on December 29, 1981. He attended elementary school and middle school in Columbia, SC, and graduated from Collins Hill High School in Suwanee, GA, in 2000. He received a B.S. in Aerospace Engineering in 2005 from the Georgia Institute of Technology. He is the recipient of the 2005 Donnell Dutton Outstanding Senior Award presented by the Georgia Tech Guggenheim School of Aerospace Engineering. In 2007, he received a M.S. in Aerospace Engineering and a graduate certificate in Management of Technology from Georgia Tech. While at Georgia Tech, he worked under the guidance of Dr. Robert Braun in the Space Systems Design Lab. He continued his education at the University of Texas at Austin under the guidance of Dr. E. Glenn Lightsey. He has completed eight rotations as a cooperative (co-op) education employee at the NASA Johnson Space Center in Houston, TX, where he has worked in the Mission Operations Directorate, the Space and Life Sciences Directorate, the Engineering Directorate, and the Constellation Program Office.

Permanent address: 11915 Stonehollow Dr., Apt. 1417
Austin, Texas 78758

This dissertation was typeset with L^AT_EX[†] by the author.

[†]L^AT_EX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T_EX Program.