AUTONOMOUS NAVIGATION FOR SPACECRAFT USING STARS, PLANETS, AND PULSARS

A Dissertation

by

STOIAN BORISSOV

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Daniele Mortari
Committee Members,	Gregory Chamitoff
	Suman Chakravorty
	Ergun Akleman
Head of Department,	Rodney D. W. Bowersox

May 2020

Major Subject: Aerospace Engineering

ABSTRACT

Autonomous navigation capabilities for spacecraft are useful for decreasing operational costs, increasing the scope of missions, and increasing the robustness of spacecraft. This dissertation aims to present and fully explain multiple autonomous navigation methods that may be used by spacecraft orbiting Earth or on interplanetary (and potentially interstellar) missions. A brief history of each navigational technique will be given, with a summary of state-of-the-art developments, as well as results from implementing the navigational technique in hardware or software simulations. The results of a simulated Orion mission traversing cis-lunar space while implementing optical navigation techniques will be shown along with a detailed explanation of the simulation architecture. An analysis of how existing navigational techniques can be improved or superseded by pulsar-based navigation methods will also be presented. Finally, conclusions and recommendations for future work will be made.

I dedicate this dissertation to all autonomous navigators

who wander in search of truth.

Keep searching.

ACKNOWLEDGEMENTS

I owe much to my advisor Dr Daniele Mortari. He is the one who provided me with the opportunity to work on a plethora of fascinating research topics at the cutting edge of human enlightenment. I must thank him for providing guidance while also having the patience to let me figure things out on my own. He gave me the freedom to find my own path, even when at times that pathfinding was difficult or painful. More than just an advisor he's a great chef, connoisseur of coffee, and has become a trusted friend. Daniele, graci.

I must also thank The Committee for their support and guidance as I've worked on my dissertation. Also, a special thanks to Dr Gregory Chamitoff for his support and for giving me opportunities to learn about human spaceflight, and Dr Vikram Kinra for providing me opportunities to become a mentor for so many other students though the Graduate Teaching Fellows Program. These experiences have made a profound impact on my career aspirations.

I also want to thank all my friends and colleagues in the Department of Aerospace Engineering. I've enjoyed sharing ideas through stimulating conversation and staying up late wired on energy drinks debating the finer nuances of our latest confounding assignment from our professors and advisors. The short list is Hunter, Carl, Robert, Anthony, Francesco, Allen, Kristin, Ben, Mauricio, Derek, Bharat, Nathan, Vinicius, Farid, and it goes on ... Thank you for the coffee, beer, couches/beds to crash on, and for being there to listen when I needed someone to talk to.

My students also deserve acknowledgement. I want to thank my students in

the classroom for supporting me in the beginning as I was giving my first lectures and later on as well, when I began making big changes to the coursework and giving everyone tons of homework. Your effort, feedback, and willingness to work with me has been instrumental in helping me grow as a mentor and in shaping my ideas of what quality education should be. I also want to thank my research students, whose self-motivated exploration of our X-Nav research topics helped me grow again as a mentor, and also as a researcher. Thank you for putting your faith in me, and I hope that I have been able to provide you with opportunities to learn and opened doors for some of you to build your futures.

I want to thank my parents for providing me with all the resources and mental tools to succeed in an environment which requires critical thinking, hard work, and a healthy disrespect for convention. Specifically, I want to thank my dad, Roumen, for teaching me from a young age to question my own ideas and assumptions and to prioritize integrity and the search for truth over the desire to always be on top. And, I want to thank my mom, Milena, for teaching me to question the rules and assumptions of others and think for myself while also finding common ground when it is there.

I must also acknowledge my virtual mentors. While we've never met, and you have no idea who I am, I truly appreciate the work you do and am grateful to you for sharing your ideas with the world. From Sean Plott, to Josh Waitzkin, Tim Ferris, Simon Sinek, and YouTubers Derek Muller from Veritasium, CGP Grey, and Michael Stevens from Vsauce, as well as the folks behind so many of the educational and creative online channels like Kurzgesagt and Sixty Symbols. You have taught me so much and introduced me to so many other great mentors, thinkers, and writers from both the past and present. You represent what I find most beautiful about this modern age of internet communications and I strive to join your ranks one day. Keep growing and sharing!

Finally, I would like to acknowledge any and all individuals that have helped me grow over my years at A&M and in College Station. This includes, not only those that have taught me and shared their wisdom with me, but also those that have challenged me, and forced me to rise to the occasion to meet those challenges.

Stoian Borissov
College Station, TX
March, 13th 2020

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Professor Daniele Mortari acting as advisor and Professors Gregory Chamitoff and Suman Chakravorty of the Department of Aerospace Engineering and Professor Ergun Akleman of the Department of Visualization. Initial algorithm development for OpNav performed by graduate researcher Francesco DeDilectis.

The following organizations have contributed to the work presented in this dissertation.

Orion OpNav Team, Group EG2 of Engineering Directorate, NASA Johnson Space Center

Orion OpNav development efforts led by Chris D'Souza at JSC. Camera calibration and testing work performed in Electro-Optics Lab with aid of fellow Coop Interns under the supervision of Steven Lockhart. Trajectory simulations for the simulated EDGE images provided by Emily Kollin. Camera calibration, characterization, and sensitivity analysis performed by Bryce Coughran, Emily Zimovan, and Thomas Leps. Camera control developed by Ronney Lovelace. Algorithm testing assisted by Peter Schulte. Moon images from the ISS provided by astronaut Terry Virts. Development of EDGE supported by Frank Graffagnino. Additional support, image processing and navigation algorithms provided by Professor John Christian of WVU.

All other work conducted at JSC for the dissertation was completed by the student independently.

Aggie Research Programs, Texas A&M

X-Nav research supported by the Aggie Research Leaders and Scholars Programs through contributions to research efforts by undergraduate students at Texas A&M. Initial studies into pulsar glitches done by Grayson Bridges. Ambiguity resolution algorithms and initial investigation into pulsar navigation explored by Samuel McConnell, Michael McCarthy, and Anthony Young. Background research on pulsar signal processing performed by Victoria Write and William Vlasak. Gerardo Barillas investigated implementing X-Nav for interplanetary mission simulated in SPICE.

All other work conducted at Texas A&M for the dissertation was completed by the student independently.

Funding Sources

Graduate research work was supported by the Department of Aerospace Engineering at Texas A&M, by NASA-JSC Contract NNX14AK47A, and by the Graduate Teaching Fellowship from Texas A&M University.

OpNav algorithm development and testing at JSC was supported by a stipend through the NASA COOP/Intern program.

Travel funds for presenting at the 2018 AAS/AIAA Spaceflight Mechanics Meeting provided by the AAS Breakwell Student Travel Award.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

NOMENCLATURE

To facilitate understanding the development behind X-ray Pulsar-Based Navigation, in particular the ambiguity resolution algorithm presented in Chapter VI, the following variables and terms are introduced:

Nomenclature

c	Speed of light
$\mathbf{p}_0$	Initial estimated observer position
τ_0	Initial estimated observer time
δ_τ	Clock drift / time uncertainty
δ_r	Scalar position uncertainty
σ_t	Timing noise associated with TOA
N	Total number of observed pulsars
n	Pulsar counter, $n \in [1, N]$
M_n	Number of pulse waves in search space associated with P_n
i	Pulse wave counter, $i \in [1, M_n]$
P_n	Pulsar ID
T_n	Pulsar period
ϕ_n	Pulsar phase
$\hat{\mathbf{n}}_n$	Pulse direction
$P_{n,i}$	Pulse wave ID

Nomenclature Cont.

$TOA_{ref,n,i}$	Time of arrival at reference point for pulse wave $P_{n,i}$
$TOA_{obs,n,i}$	Time of arrival at observer for pulse wave $P_{n,i}$
$\Delta TOA_{obs,1,b}$	Difference in TOA_{obs} between pulse waves $P_{1,i}$ and $P_{b,i}$, for $b \neq 1$
$\Delta\phi_{1,b}$	Difference in phase between pulsars P_1 and P_b, for $b \neq 1$
t_w	Working time, $t_w = TOA_{obs,1,i}$
$\lambda_{a,b}$	Minimal lattice distance for pulsars P_a and P_b
$\lambda_{\max}$	Largest minimal lattice distance, $\lambda_{\max} = \lambda_{1,2}$
A	Overlapping solution area between multiple pulse waves
(P_1, P_2)	Pulsar pair with Largest minimal lattice distance, $T_1 > T_2$
$\mathbf{I}_k$	Candidate solution set, set of wave IDs that simultaneously overlap
$\mathbf{S}_k$	Candidate solution: $\mathbf{S}_k = (x, y, t)_k$
$\mathbf{I}_{\min}$	Minimal solution interval pair
$[T, D]$	Solution interval
$[T_{\min}, D_{\min}]$	Minimal solution interval

Definitions and Terms

Solution Interval: Gap in time or distance between two possible solutions:

Solution Area: Area created by overlapping pulse waves of finite width $2c\sigma_t$

Intersection: Exact intersection between two pulse waves at any time or three pulse waves at a unique intersection time

Conjunction: Intersection or close approach between three or more pulse waves

Overlapping Solution Area: The point or convex polygon created when waves of finite width $2c\sigma_t$ intersect

CHAPTER I

INTRODUCTION AND MOTIVATION BEHIND WORK IN AUTONOMOUS SPACECRAFT NAVIGATION

We begin with brief introduction to the field of study called *Navigation* and present justification for the work being done in this field. The reader should get a sense of why the topic of autonomous spacecraft navigation is important and why research in this field is valuable.

Introduction to the Field of Study: Navigation

The problem of precise and accurate navigation is a rich field that has been of interest to explorers, engineers, merchants, and lost travelers for thousands of years. It has its applications in nearly any industry that involves moving material, payloads, people, and goods from one place to another. For the purposes of this document, we are particularly interested in the navigation of spacecraft. First, it pays to understand what is meant by the term *navigation*. Here are several dictionary definitions:

- From Merriam-Webster

  navigation (*noun*)

  1. : The act or practice of navigating
  2. : The science of getting ships, aircraft, or spacecraft from pace to place. *Especially* : the method of determining position, course, and distance traveled

- From the Oxford English Dictionary

  navigation (*noun*)

  1. : The process or activity of accurately ascertaining one's position and planning and following a route.
  2. : The passage of ships
  3. : *Computing* : The action of moving around a website, the Internet, etc.

These are fine definitions, but for the sake of clarity, let us refine the definition and specify what is not meant. In some fields, the term *navigation* might be more closely associated with the disciplines of path planning, obstacle avoidance, and mapping. However, that is not how the term will be used in this document. Instead, we take *navigation* to indicate methods used to better understand one's physical place in the universe:

> In this document the term ***navigation*** will be used to mean localization and estimation of the *pose* of a spacecraft at a specific moment in time.

Where *pose* is the position and attitude, or orientation, of a body with respect to some reference frame. Thus, navigation implies the time-knowledge of both, orientation and position. Though the focus of this document will be the development and implementation of position estimation algorithms, position estimation is tightly linked with attitude estimation, and therefore some discussion will also be dedicated to the latter. Attitude estimation is a process which is well understood and has been implemented on spacecraft for decades using devices known as star trackers which use the fixed stars as orientation markers. The complexity of position estimation has made it a more elusive target, one which cannot be obtained by just looking at the stars, and one which warrants further exploration and development.

What is Autonomous Navigation?

The term *autonomous* means more than just *automated.* Yes, autonomous navigation can imply the use of computers and algorithms to perform navigational estimates. More importantly, however, the problem of autonomous navigation deals with a craft, with crew or without, or machine navigating itself and learning its own position, sometimes with, and sometimes without, external aids. To successfully navigate, a craft must be able to estimate its state with respect to some external reference. More specifically, it must be able to navigate with respect to a *reference frame.* For the purposes of this document a reference frame is a set of spatial directions (which can evolve over time) that are generally defined about some origin (which can also move over time). A craft selects a frame and then uses the directions provided by that frame to describe its position. The position may be reported with respect to the origin of that frame or with respect to some other object or reference point in space. To do this, a spacecraft needs two things: 1) a world it can measure and observe and 2) instruments that will perform the measurements and observations. A world rich with information and navigational markers is no good to a craft which is blind, while conversely even a spacecraft with the most expensive state-of-the-art suite of instruments is lost if there is nothing in its surrounding environment for it to sense and observe.

Motivation Behind Work and The General Concern

That's a sufficient introduction to the topic of navigation. The next question is: Why is the topic of autonomous navigation relevant in the context of spacecraft? There are several factors which inform and drive the development of autonomous spacecraft navigation. Briefly, these include:

- A concern for astronaut safety
- A need for more robust navigational systems that can tolerate a loss of communications with Earth-based ground stations
- An expansion of the operational envelope for rendezvous and proximity operations far from Earth
- A reduction in the workload for ground-based satellite operators and systems which must control spacecraft maneuvers from Earth
- Greater efficiency of spacecraft operations that comes from being able to act without waiting for input from a ground-station operator

Taken together, these factors would dramatically increase the operational envelope of spacecraft, allow for more efficient operations, increase robustness of on-board systems, and eventually support a significantly larger presence of humans and their machines in space. A fully capable suite of autonomous navigation systems would form part of the foundation that is needed for a far-flung interplanetary human presence in space and eventually, a space-based economy. So, the long-term incentives driving the development of autonomous spacecraft navigational systems is strong. A subset of these incentives however presents a pressing need for immediate improvement of spacecraft navigational capabilities. Those two incentives are the need for backup navigation in the case of loss of communications with Earth and the concern for astronaut safety. To understand the significance of these reasons, one must first understand the current methods employed on existing spacecraft navigational systems.

Existing Methods in Spacecraft Navigation

Currently, the majority of spacecraft rely heavily on communication with Earth for navigational purposes. This means that most spacecraft don't actually know their own position or velocity in space, both of which are key pieces of information when doing maneuvers. This information is collected and monitored from Earth via radar and ground stations and sent to spacecraft as periodic navigational updates. The spacecraft might be able to propagate this position and velocity information forward in time for a small duration, but unless there is some way to get an updated navigational fix, the spacecraft will become lost. The exceptions to this rule are spacecraft which are specifically designed to implement and test autonomous navigation capabilities and spacecraft which perform relative navigation during rendezvous operations. These spacecraft have means of updating their position and velocity estimates without relying on communication with Earth. However, even in these cases, having the additional navigational information sent from Earth can act as a sanity check and help increase the confidence of executing certain maneuvers. If a spacecraft does not know its position and velocity correctly, then it may perform an orbital maneuver at the wrong time or incorrectly apply a thrust, both of which could jeopardize the survival of a mission and lead to loss of the spacecraft or at least degradation in performance and loss of capabilities.

Impact of Loss of Communications with Earth on Spacecraft Navigation Capabilities

Clearly, communications with Earth play a significant role in spacecraft navigation. Thus, a spacecraft's loss of communications with Earth, either due to a technical fault or occultation by another body, can pose a significant risk to the safety of a mission. In order to mitigate this risk autonomous navigational capabilities can be implemented on-board which allow a spacecraft to either return to a position where it can reestablish communication with Earth or at the very least be able to maneuver correctly and thus keep itself alive long enough for engineers on Earth or crew on board to attempt to fix the problem. If a spacecraft can buy itself enough time, it may be possible to salvage the mission. Some scenarios where communication loss might occur include:

- Mechanical fault due to vibrational shock after maneuver, structural reconfiguration, deployment of a system, or stage separation
- Damage from debris impact or other mechanical malfunction
- Loss of power
- Software glitch or single event upset (SEU) due to cosmic radiation
- Regularly occurring interruptions in communication as a spacecraft orbiting

another celestial body is in opposition with respect to the Earth and the celestial body is occulting communication signals.

While it is prudent to invest effort in reducing the chance of some of these occurrences, one can never guarantee they will never occur, and in fact might be a part of nominal mission operations, such as orbiters around other planets. For a system at risk of communications loss autonomous navigational system adds a degree of fault tolerance which reduces risk to the mission. Even for spacecraft which routinely experience periodic interruption of communications with Earth, having an autonomous navigation system on board expands the operational capabilities of the spacecraft allowing it to perform maneuvers even while in the "shadow" of a celestial body and direct communication is not possible.

NASA's Concern with Astronaut Safety

With the presence of crew on board a spacecraft the need for more robust navigational capabilities is even greater than for uncrewed missions. Having autonomous navigational capabilities frees up load on the crew and provide them more time for other mission critical work. It can also expand the operational scope of the mission as the crew can act autonously with no need for feedback from Earth. However, most importantly, having an autonomous navigation capability on board a crewed spacecraft will give an additional margin of safety to the crew on-board. Like in the days of the Apollo program, and specifically Apollo 13, astronauts on NASA's newly developed Orion spacecraft will have the ability to manually take angle measurements with devices such as sextants, and this can be used for estimating position and direction of travel. However, in the case of communication loss, or an accident which may incapacitate crew members, there is need for a system which can intervene and guide the spacecraft safely for as long as possible. This is a major driver for the development of autonomous navigation and is one of the main motivations that has spurred the development and testing of the algorithms described in this document.

Overall Scope of Work and Concern

The position estimation algorithms described in this document are meant to augment the navigational capabilities of spacecraft which are operating either in deep space or medium to high orbit around an ellipsoidal celestial body. For the purposes of this document, "deep space" means an operational environment for a spacecraft where all celestial bodies excluding the primary star of a system appear as point sources of light, i.e. as bright stars. "Medium" and "high orbit" are used as relative terms to mean distances between a spacecraft and a primary celestial body at which the primary appears as an extended body, specifically an ellipsoid, that can be seen in its entirety within the field of view of an optical camera on board the spacecraft.

Deep space missions include interplanetary missions from the Earth to Mars where a spacecraft would be far enough from both the Earth and Mars at a given moment that each would appear as just a point source of light. In this sort of operating regime, no information pertaining to the geometry of a celestial body can be used.

Medium and high orbit space include the space between the Earth and Moon, or cis-lunar space, which includes the geosynchronous belt as well. This operational regime is relevant to any missions to the Moon and back. However, it is not mandatory to only be considering orbital space around Earth. The space around any moon or planet may be considered.

Limitations of Scope

The work in this document does not consider the navigational problems and challenges associated with rendezvous between spacecraft, proximity operations, low orbit navigation where the field of view of a camera is insufficient to capture the entirety of the primary body, and orbits around irregular bodies which significantly deviate from an ellipsoidal shape, such as asteroids. While the techniques described in this document could be used in these scenarios, they are not well suited for these operational modes, and there are other methods which better suit the needs of spacecraft operating in these conditions.

Optical Navigation as a Method of Autonomous Navigations

In this section of the introduction the general technique of OpNav will be described along with the motivation behind its development. OpNav is an abbreviation which stands for Optical Navigation. It implies the use of cameras to take either pictures or video, in any spectrum of interest, of markers or bodies which can aid in navigation, or more precisely, aid in the estimation of a spacecraft's position, orientation, velocity, and time. The general term for such data collection using cameras is *imaging*. An example of such imaging can be seen in Figure I.1 with the captured image defined in the camera coordinate frame. OpNav is not the only means of autonomous navigation, however it is the means that was chosen for development in addressing NASA's concern for astronaut safety for future crew on board the Orion spacecraft on return from a mission that takes them to or around the Moon.

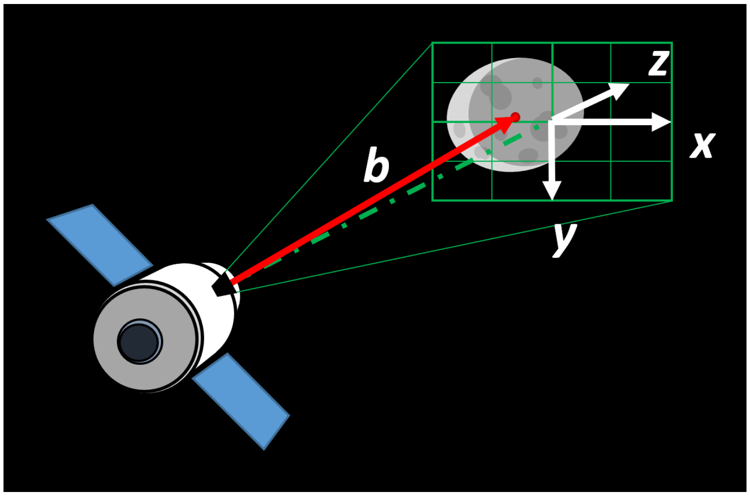


Figure I.1: Imaging of ellipsoidal body by spacecraft. Image is defined using camera coordinate frame.

Specific Concern and Motivation behind Cis-Lunar Autonomous Navigation for Orion

With the retirement of the Space Shuttle, NASA is looking for new and alternative means for transporting astronauts to orbit and beyond. An integral part of this effort has been the development of the Orion Space Capsule. The capsule, developed as a joint effort between NASA and Lockheed Martin, will be responsible for bringing crew members up to the ISS and other orbiting platforms of the future and eventually bringing astronauts to the Moon and Mars. One of the first milestones which Orion must march past is to repeat what the Apollo program did over fifty years ago – fly around the Moon and return to Earth safely. Demonstrating this capability is one of the goals of the Artemis 1 mission (formerly Exploration Mission-1, or EM-1), currently slated to launch no earlier than November 2021. While the mission will be uncrewed, it will test a lot of the same systems which will be needed for a crewed flight and will include a life support system and seats with mannequins strapped in that will have sensors on them that measure radiation exposure. Orion is planned to spend several days in retrograde orbit around the Moon before maneuvering and starting its return trajectory back towards Earth. While navigation in space always carries risk with it, the return to Earth is particularly dangerous in part due to the fact that there is limited fuel remaining for maneuvering and adjusting the trajectory. Orion must enter a relatively narrow trajectory, called a return path corridor, in order to ensure that it can safely return to Earth at the correct entry angle. If the return path corridor is not targeted correctly, for example, the angle of Orion's trajectory is too steep with respect to the Earth's surface, Orion will burn up in the atmosphere, killing everyone on board. If the angle to too shallow, there is a risk that Orion might bounce off the Earth's atmosphere, much like a stone bouncing off the surface of a pond, and be sent back into outer space, where it might not have enough fuel to properly maneuver back and thus, again, resulting in loss of crew. So, getting Orion on that return path corridor is very important!

In order to hit that return path corridor, Orion's maneuvering thruster must be fired at the correct time, in the correct direction, at the correct point along its orbit around the Moon. Additional corrective maneuvers can be done to fix the trajectory but that can only be relied on only up to a certain point, after which there is nothing that can be done to effectively redirect Orion. In order to perform these maneuvers well, Orion must know its position in space so that it knows when to initiate thruster firing. Generally, a spacecraft knows its position in space by communicating with ground stations on Earth which monitor its progress as it moves across the heavens and report back to it navigational information. This means that a spacecraft's navigational system is highly dependent on the communications system. If there happens to be a loss of communications for any reason, such as a computer glitch, or power fluctuation, a spacecraft can temporary, or permanently, get lost. If this were to happen to Orion while around the Moon, Orion could be in very big trouble and unable to properly maneuver its way back home. This is one of the main incentives which drives the development of an autonomous navigation system. If something were to happen with Orion's communications while around the Moon or in cis-lunar space, then Orion could potentially still get home safely using its autonomous navigation system.

To meet these navigational needs, NASA engineers decided that an optical navigation system using a camera that takes pictures in the visible light spectrum is appropriate for the needs of Orion. Such a system would work by taking pictures of either the Moon or the Earth and attempt to locate the spacecraft relative to one of the observed bodies. These types of systems have flight heritage, and are well suited for cases where planets and moons (and the Sun) appear as extended objects, as shown in Figure I.2 for a hypothetical interplanetary mission from Earth to Mars. OpNav is particularly well suited for cis-lunar space, since both the Earth and Moon appear as extended object ellipsoids allowing for relatively accurate distance estimation. In this environment other planetary bodies, such as Mars, Venus, and Jupiter appear as point sources of light (unless using magnification) and thus make relative distance estimation inaccurate at best and completely impossible at worst.

The General Problem of OpNav

The general problem of OpNav can be summarized as the task of imaging an extended object celestial body and estimating its apparent center and distance to the observer. Combined with orientation, or attitude, information, this is enough to form a position estimate with respect to the observed body. If the observed body has a known position with respect to some other reference, such as the Earth, for example, then the observer can translate their position estimate with respect to that reference. Thus, in OpNav accurate position estimation requires accurate centroiding and size estimation. There are multiple challenges associated with this task, such as sources of error coming from the hardware, i.e. imaging optics and attitude sensors,

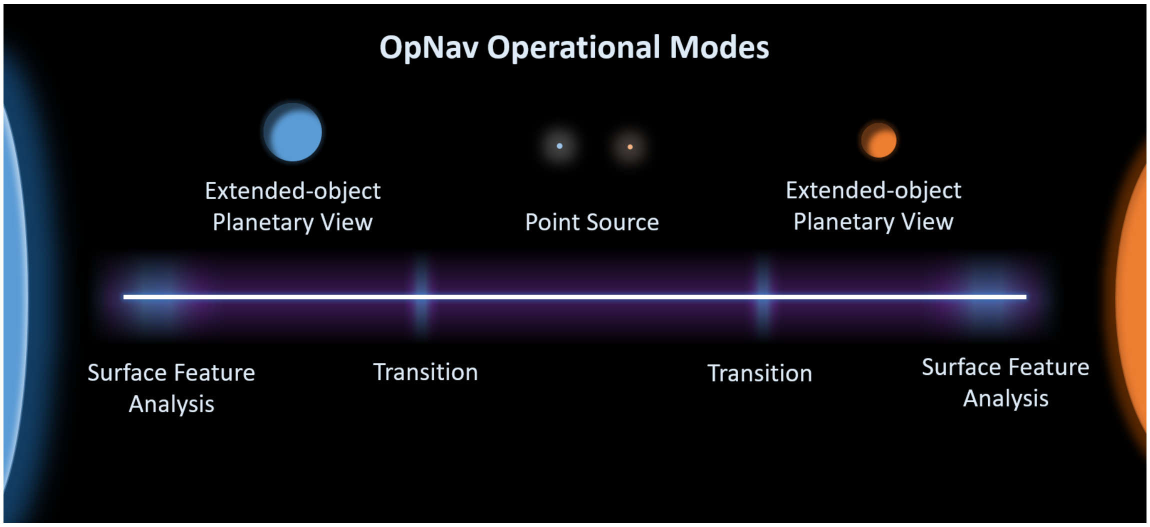


Figure I.2: Different modes of OpNav for spacecraft traveling from Earth to Mars. As a spacecraft leaves one celestial body an travels towards the other, the appearance of the bodies will vary: from taking up nearly half of their field of view and with multiple surface features visible, to appearing as point sources of light.

as well as algorithmic biases that must be accounted for in order to perform OpNav well. These challenges will be discussed in detail in the following chapters along with methods for resolving them.

X-Ray Pulsar-Based Navigation as a Method of Autonomous Navigations

X-ray pulsar-based navigation is a promising new technology in its nascent stages of development that could potentially solve a lot of the needs associated with autonomous navigation. Like OpNav, X-Ray pulsar-based navigation, X-Nav for short, is an autonomous navigation solution which is suitable for spacecraft in mid to high orbits. As a simplistic analogy, X-Nav can be thought of as a galactic-scale version of Earth based Global Navigation Satellite Systems (GNSS) used for navigation. Pulsars are neutron stars with powerful magnetic fields which produce radiative jets which shoot out of the stars' magnetic poles. These magnetic poles are inclined with respect to the stars' rotational axis, and thus the star behaves like a strobe light, or lighthouse. This is the origin of the "pulsing" effect which gives pulsars their name. This is visualized as an artist rendering in Figure I.3.

Astronomers first discovered pulsars in the '60's and have been cataloging them ever since. Pulsar-based navigation can only work with these previously observed and cataloged pulsars. To date there are several thousand pulsars in the Milky Way that have had their locations, periods, phases, pulse profiles, and various other parameters cataloged. While there are slight variations between these pulsars, it is generally true that a pulsar's spin is incredibly stable, which means its signal is very steady and predictable. The stability of the signal coming from some pulsars rivals that of atomic clocks and this is where the utility of pulsars as navigational aids comes into play. Just as satellite navigation systems, such as GPS, which most modern phones can use to synchronize their own clock and estimate their position, theoretically, so too can pulsars be used in the same fashion by spacecraft.

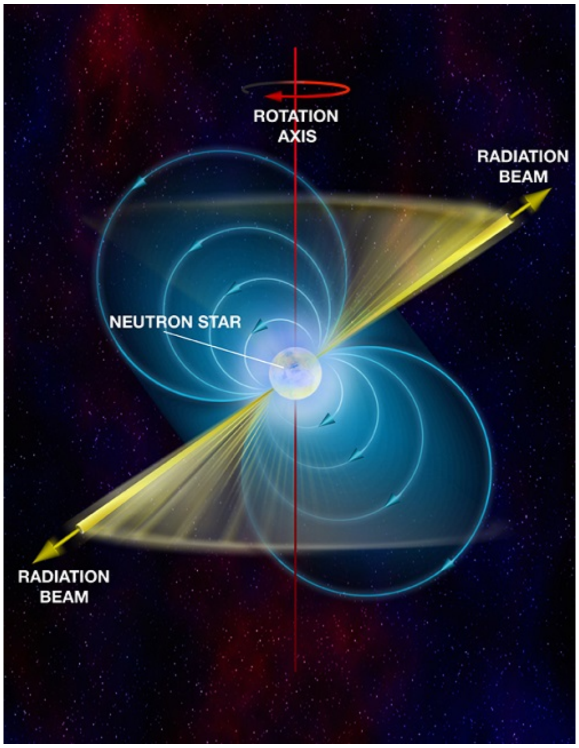


Figure I.3: Pulsar radiating energy outward (yellow) along its magnetic poles as it spins. The white center is the neutron star with magnetic field lines shown in blue. Credit: B. Saxton, NRAO/AUI/NSF

Since pulsar signals are so regular, it is known to within a very tight precision exactly when their pulses are supposed to arrive at a given location in space. Generally, a single period from a pulsar signal looks like the signal shown in Figure I.4. Pulsar signal often have a single primary peak and a secondary dimmer peak. This is due to how the magnetic poles of the star are aligned with respect to the observer. The magnetic poles are generally neither in line with the rotational axis, nor are they orthogonal to the rotational axis. Thus, if one pole perfectly aligns with the observer and produces a bright pulse, once the star completes a half-rotation, the opposing pole will be pointing roughly in the direction of the observer, but not directly at them. This is what produces the secondary dimmer pulse.

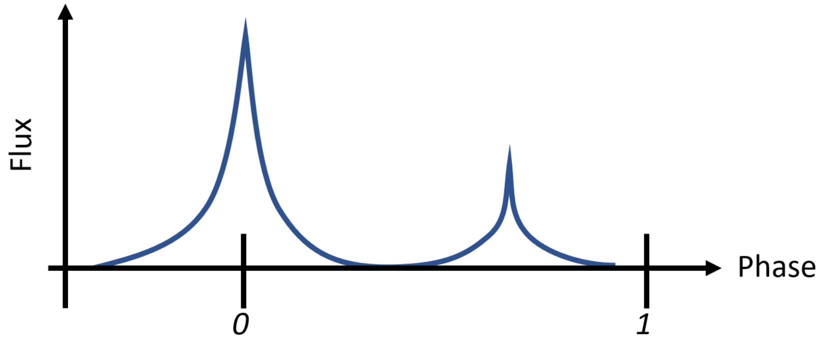


Figure I.4: An approximation of what a single period from a pulse chain looks like. Note that the phase is generally aligned with the peak of the signal.

To use the pulse signal, an observer must be able to assign an arrival time for the peak. If the signal is resolved and an observer knows their position, they can correct any drift in their clock. Conversely, if an observer trusts their clock, but is uncertain of their position in space, based off of the difference in the time of arrival between an expected and observed pulse, an observer can update their position estimate. If one is certain of their clock, then it is theoretically possible to determine position using pulsar signals coming from three distinct direction. If an observer is uncertain of their clock and their position, then it is possible to estimate both position and clock drift by observing four pulsars. This can be done anywhere in space. Since pulsars are stars which are distributed throughout the galaxy, there is no restriction to where this method can be used, though it is best suited for use when far away from a large celestial body, so that the body does not interfere with the relatively dim signal coming from the pulsar.

Specific Concern and Motivation Behind the Development of X-Nav

X-Nav is potentially a useful tool to develop since it offers an additional method of navigation for spacecraft. Generally, the more navigational methods available to a craft, the more robust the overall navigational system is. Faults can be detected and results can be corroborated by multiple independent methods and systems. It also serves as another additional backup in the case of failure of communications or failure of other navigational systems, thus relieving some of NASA's concerns regarding safety of crew and spacecraft. However, these are not the main drivers behind the development of X-Nav.

The largest drivers behind the development of X-Nav are, first, the universal availability of pulsars as navigational beacons and second, the extreme stability of their signals. If the technical challenges of development can be overcome, then X-Nav could become a universal navigational system which is capable of not only providing positions estimates for spacecraft, but also correct clock drift, and even estimate spacecraft velocity. Plus, in a deep space scenario, where perhaps no other celestial bodies are readily visible for navigation, pulsars will always be available for observation. Thus X-Nav could potentially be a universally available, maximally accurate navigational and timing system. Before X-Nav fulfills that promise however, there are significant technical challenges which must be overcome.

The General Problem of X-Nav

One of the largest technical hurdles standing in the way of implementing X-Nav is the development of the capability to resolve signals from known pulsars when observing them from a space-borne platform. Resolving and processing the signals coming in from multiple pulsars can be broken down into roughly three separate steps:

1. Recognizing which pulsars are being observed: The first step in using a pulsar for navigational purposes is to know which pulsar is being observed. Each pulsar's signal can act as unique fingerprint, so if the signal can be resolved, it would be possible to use it to identify the pulsar. However, resolving the signal requires observing the pulsar for an extended time and using *a priori* knowledge of the pulsars period. Another way to identify a pulsar would be to use its location in the night sky. Pulsars are stars, and thus their angular position in the night sky does not change as an observer moves around the solar system. If an observer can estimate the attitude of the camera observing the pulsars, then it is possible to know which pulsar is being observed. Once a signal can be associated with a cataloged pulsar then all of the important signal characteristic for that pulsar can be pulled up such as period, phase, brightness, location, and so on. This information is key for X-Nav.

2. Collecting sufficient light to resolve pulses: Resolving a pulse is the process of collecting enough light from in incoming pulsar in order to be able to re-create the pulse waveform and estimate the arrival time of the signal peak. There are several challenges to this process.

Pulse signals are incredibly dim. The radiation beams of different pulsars emit in different frequency bands. Some pulsars have peak power emission in the radio band, while others might have peak power in higher frequencies including visible, X-ray, and gamma. Some have narrow-band emissions while others, such as the pulsar at the center of the Crab Nebula, emit broadband. Pulsars that emit large amount of energy in the radio band are relatively easy to observe on Earth, as long as one is equipped with a sufficiently large radio telescope. However, such measurement systems are not practical for space-borne application, and thus astronomers and engineers have turned to pulsars which emit more of the their energy in higher frequency bands, particularly X-rays, when looking for pulsars from space. Receivers for such signals can have a smaller form factor, and thus are more practical for observations from space. X-ray pulsars tend to also be millisecond pulsar, which, due to a mass-accretion process, have spun up their rotational rates to the point of having periods on the order of one to tens of milliseconds. These X-ray millisecond pulsars are the prime candidates for a pulsar based navigational system, hence the name X-Nav. However, using X-ray pulsars comes with the added challenge that the X-ray bands generally have a lower power density than other lower frequencies, making the signals even harder to detect. The signals are dim enough that they are strongly affected by shot noise and have to be modeled as a chain of individual photons which arrive according to a non-homogeneous Poisson process. A signal must be observed for long enough that sufficient statistical data is collected to be able to reproduce the waveform and estimate the arrival time of the peak. Franzen et al. [1] demonstrate how the waveform of the Crab Pulsar, PSR B0531+21, the pulsar at the center of the Crab Nebula, can be reconstructed with a sufficient amount of light collected from observations.

To gain enough statistics to reconstruct one period of the signal, an observer may have to collect hours or days worth of data — thousands of periods — just to be able to estimate the time of the first peak to arrive after data collection began. This is possible since the periods of the pulsars are already known, so a signal processing technique known as epoch folding can be used to overlay data from consecutive periods onto a single period so that enough data can be aggregated to resolve the waveform and estimate a peak arrival time.

One additional complication which must be addressed is the signal dispersion which happens as the pulse travels through interstellar space. The interstellar medium acts as a refractor of light, thus light at different frequencies that might have left a pulsar at the same time will arrive at an observer at different times. Even when a narrow band of frequencies is being considered for observation, not all the light is at exactly the same frequency, so that band is discretized into channels. Over its long journey through interstellar space, the light across the various channels is dispersed, so that the peaks of each channel are no longer coherent and arrive at different times. Thus, in order to reconstruct a signal which is sufficiently bright, there must be a correction for the dispersion applied to the channels so that the peaks in each channel can all be aligned and sufficient data can be collected to stand out from the background noise.

These challenges of signal processing are not the main focus of this dissertation, however they are useful to mention since they illustrate the difficulty of precisely timing the arrival of an incoming signal. The mechanisms explained here are the reason behind the timing noise which is present when collecting pulsar observations. The estimation and knowledge of this timing noise is critical for the implementation of a pulsar-based position estimation algorithm.

3. Differentiating between pulses and eliminating ambiguous solutions: The final component, and the focus of the developments in X-Nav discussed in this dissertation, is the method of distinguishing incoming pulses coming from the same pulsar. Pulsar signals are not modulated over multiple periods, meaning that all pulses look identical. The goal of X-Nav is to use the timing of an arriving pulse to help estimate the position of an observer by comparing when it was seen by the observer to when it was expected to arrive at some reference location. However, since all pulses from a single source look the same, it may be impossible to determine which pulse in a sequence was observed, and thus associating a correct arrival time to a position estimate may not be possible. Collecting observations from multiple different pulsars helps reduce, but not eliminate, this ambiguity. This problem, known as the ambiguity problem, will be explained in full detail and algorithms for handling it will be the focus of the X-Nav chapters of this dissertation.

CHAPTER II

OPTICAL NAVIGATION USING THE MOON AND EARTH - BACKGROUND, LITERATURE REVIEW, AND DEVELOPMENT FRAMEWORK

During the development of Orion, NASA Johnson Space Center (JSC) came to the conclusion that the necessity for a backup navigation solution for Orion warranted the development of an Optical Navigation (OpNav) system which would help astronauts get home safely in the event of an anomalous loss of communications during mission. The following section details the story of algorithm development and provides the necessary background to understand OpNav algorithm as well as context for their development.

The Story of Algorithm Development

Once it was established that an OpNav solution is desired as a backup autonomous navigation solution on board Orion, engineers at NASA JSC reached out to Dr. Daniele Mortari and his research group at Texas A&M University. Dr. Mortari already had existing connections with engineers at JSC and was well respected in the field of astrometry from his previous development of star tracking algorithms [2], [3], which are used for attitude estimation on board many spacecraft. Dr. Mortari had also previously worked on the development of a Moon-Sun sensor [4]; a system which was designed to seek out the direction towards the Sun based on the observed illumination of the Moon. Such a system would potentially be of use to a spacecraft trying to estimate its attitude. Unfortunately, the Moon-Sun Attitude sensor never saw the light of day and was not fully implemented due to insufficient precision. However, as often happens in scientific pursuits, old ideas get recycled and it was reborn as the initial foundation of what would eventually become the OpNav algorithms which were developed for Orion. At the core of the Moon-Sun sensor lied algorithms dedicated to identifying which pixels in an image belong the the limb, the outer circular arc, of the Moon.

A beautiful marriage of concepts occurred here. Dr. Mortari's work on star tracking algorithms involved heavy use of least square fitting of functions to the light profile coming from distant stars. The images which star trackers capture of stars are intentionally defocused so that the light from a star is blurred, creating a bright spot a few pixels across [5]. Least squares fitting helps provide an estimate of the apparent centroid of the light coming from a star to sub-pixel precision, something which could not be achieved as well with a star which is perfectly in focus. Thus, Dr. Mortari had a method for selecting pixels belonging to the limb of the Moon and a method of using least squares to estimate the center and size of illuminated spherical bodies. These two ideas came together to form the first version of the OpNav algorithms for Orion.

It wasn't as simple as just reusing the existing methods. Additional methods had to be added in order to improve the selection of limb pixels. Box filters, random sampling consensus (RANSAC), and SVD methods were all tried at various points to remove outlier pixel selection and improve the initial fitting of an ellipsoid to the apparent limb of the Moon. Once this initial limb fitting was completed, the secondary least squares fitting would begin based off of limb pixels which were selected after the initial fitting was completed. Since the geometry of the Moon could be approximated well by a sphere, the idea was that all limb pixels could be described, in a least-squares sense, by a single function which would describe the drop off in brightness which one sees when transitioning across the lunar limb. Starting from the interior illuminated Moon pixels and moving outward toward the limb, one would first see a relatively constant greytone level. Then, at the transition to the blackness of space at the limb, there would be a sharp drop in observed greytone after which it would be constant again. Such a transition is well described by a sigmoid function, or at least, that was believed at the time. The hope was that a single sigmoid function with a constant offset from zero, which would represent the radial distance from the estimated Moon center, would do a good job at describing the transition across the limb. Once this function was estimated using least squares, then an observer would have a precise idea of the center of the Moon as well as its apparent size, and thus use it for navigation purposes.

As testing went on, it became apparent that this method had some problems with it. It turns out that the lunar topography is not so smooth, and the various "seas" and highland areas that can be seen on its surface produce different greytone levels which prevent the accurate fitting of a single sigmoid function to the limb. What this means for least squares fitting is poor convergence with high residuals in the estimate for the apparent centroid and radius. However, a solution was proposed. If one views the gradient of an image of the Moon, instead of the original picture, the variations of the lunar surface are less apparent and the lunar surface appears dark, not too different from the space surrounding the Moon. What does stand out, however, is the limb itself which now appears as a bright arc of pixels. The transition in brightness across this arc is described by a Gaussian function, which is the derivative of a sigmoid function. When least squares fitting was performed on this gradient of the image using a Gaussian fit, residuals were significantly reduced and estimates of apparent size and center more accurate.

Another innovation which was introduced to simplify the algorithms was to embed the position estimate of the observer directly into the cost function which is minimized by the least squares process. Thus, once least squares fitting was complete, what was returned was not just an estimate of the apparent size and center of the Moon, but a direct estimate of the observer's position with respect to the Moon. This was possible to do only with the assumption that the inertial attitude of the camera-body frame was already known.

About a year or so into the development of the project however, a curve ball was thrown at Dr. Mortari's team, which by this time included the author of this dissertation. The algorithms were performing well, possibly even better than had been thought possible, however they weren't quite meeting the accuracy requirements which had been given by JSC. At this point JSC revealed to Dr. Mortari's team that there was another competing approach to OpNav which they were considering. JSC had contracted out work to two parties in order to be able to trade off the solutions they received. Dr. John Christian, of West Virginia University, had developed his own methods for estimating the apparent size and center of the Moon [6], and they were quite accurate and very efficient. A competition between the two teams ensued where each tried to improve their algorithms and squeeze out every last drop of performance, making them as robust and accurate as possible. Luckily for both teams, no effort was wasted, as the final decision from JSC was to implement both algorithms together. The algorithms developed by Dr. Christian served to provide a fast and precise initial fit of the lunar limb after which the least square algorithms developed by Dr. Mortari's team would tweak and refine the fit to try and estimate the position of the observer with the most accuracy possible.

What has been left out of the story thus far, is how the algorithms were tested, which as it turned out, was not a trivial matter. To test the accuracy of any method, there needs to be a "truth" to compare to. For the purposes of OpNav algorithm development, "truth" meant clear images of the Moon which were taken from space and with additional meta-data available that recorded where and when the image was taken. Without this extra information regarding the time and location of an observation, it would still be possible to run image processing algorithms on images of the Moon, but it would be impossible to know how accurate the resultant centroid and position estimates were. Thus, a significant effort had to be made to acquire or create truth images for testing and developing the OpNav algorithms. Though a few existing images of the Moon were available from some missions, the team ended up relying heavily on simulation work. A virtual simulation environment using NASA's EDGE rendering software and ephemeris data for the Moon was created and set up in JSC's Electro-Optics Lab. Though the simulation environment allowed precise control of observer position and observation time, it was not enough to just simulate how the Moon would appear to a perfect pin-hole camera observer. In order to produce synthetic images which were representative of what an actual camera would see in space, a camera was mounted in the lab which captured images projected on a high resolution 4K display. Thus any lens distortion, aberration, and thermal camera noise which one would expect to have from a real camera could be included in the synthetic images. This work was done largely during the summer of 2014.

The following summer, Steve Lockheart, a technical lead and optics expert that had been working with Dr. Mortari and Dr. Christian was able to patch a special request through to Colonel Terry Virts. It was difficult to get it through, but well worth it, because at that time Terry Virts was an active astronaut on board the ISS. The request was for a series of images to be taken of the Moon from the various viewing ports on board the ISS. Initially, the zenith facing window was used, but due to the protective micrometeorite shielding on the viewing ports, this produced images with poor optical quality. Terry knew these images would not serve the needs of OpNav development. Steve was awoken by an unexpected phone call in the middle of the night; Terry was on the other line, calling from the ISS! Terry wanted to use the nadir facing Cupola for imaging instead, due it's superior clarity. With Steve's blessing, and armed with a hand-held Nikon camera and a laptop, Terry began taking images from the ISS. However, using the Cupola introduced a new set of issues. Terry had very little time to capture the images, since he had to use the side windows of the Cupola to take pictures of the Moon, and he only had minutes available for viewing before the Moon was hidden from view by the Earth. Nevertheless, Terry captured the images of the Moon that were needed for verification of the algorithms. Along with a timestamp and active tracking of the ISS available through Two Line

Elements (TLEs) the exact time and position from which the images were taken was known. The team had its truth data and testing using real images from space could commence.

Testing was a success. Using real images of the Moon, taken from space, it was verified that the position with respect to the Moon could be estimated with the desired precision for providing a safe, autonomous, backup, navigation solution for crew members of Orion returning from the Moon. The algorithms, having been verified in the high-level MATLAB code in which they were written, then had to be converted to C and tested on NASA's Core Flight System (cFS) platform which simulates avionics and controls systems of spacecraft. This work has been ongoing as of the writing of this document. In August of 2019 NASA awarded the Orion Optical Navigation Team the Group Achievement Award "*For sustained superior performance in developing a new technology to support the safety critical navigation backup system for the Orion EM-1 mission.*" That is the story behind the development of the OpNav algorithms for Orion in a nutshell. What follows is a complete detailed technical breakdown of the story just told.

Background on the OpNav Position Estimation Problem

First and foremost, OpNav requires an observer to be able to capture an image of a target body and then identify its location and distance with respect to the observer. The image processing techniques covered in this dissertation are tailored to interplanetary navigation scenarios. These include missions between planets and missions between moons and planets, where the distances from observer to target body are at least several thousand kilometers and any surface features are ignored. Most large natural celestial bodies have sufficient gravitation to pull their mass into an approximately ellipsoidal volume with a readily defined geometric center. When imaging a body, one can compare what is observed to *a priori* knowledge of the body geometry to estimate their distance from its geometric center. This is the basic principal behind OpNav when implementing it as an interplanetary navigation aid.

Relative and Absolute Position Estimate

For navigation purposes, it is useful to know the position of a spacecraft relative to the center of the Earth. However, in the case of Moon-based Op-Nav, before an observer can estimate their position with respect to the Earth, it is first necessary to know their position relative to the Moon. The Moon, being a celestial object which has been observed and tracked for thousands of years, has it's position over time with respect to the Earth, cataloged with high precision. Therefore, if a spacecraft can accurately estimate it's position with respect to the Moon, it is a relatively simple question of translating that position estimate to be with respect to the Earth. As long as the date and time of the observations used in forming the position estimate are well known, then the distance and direction from the center of the Moon to the center of the Earth is known. As shown in Figure II.1, vector addition is all that's needed after that in order to determine the position of a spacecraft with respect the Earth and get and absolute position estimate in an inertial frame.

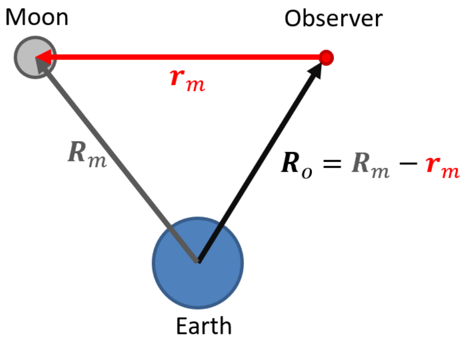


Figure II.1: Observer position relative to Moon can be translated to position relative to Earth.

A relative position estimate with respect to the Moon, or any other target body, requires knowledge of the distance to the target as well as the orientation, or attitude, of the spacecraft with respect to some reference frame. What reference frame is used for expressing the spacecraft's attitude depends entirely on what sort of position and attitude estimate is useful to engineers and mission operators.

Though attitude estimation is not the focus of this dissertation, it is a necessary component of position estimation, and therefore at least a brief overview is necessary. One of the most accurate and universally available methods of attitude estimation is the use of star catalogs and star identification. Objects which are observed by a camera in a scene can initially only be localized, i.e. have their position expressed, using the coordinate frame of the camera. If the attitude of the camera is known, then a rotation matrix C (or other mathematical tool) can be used to express the orientation of the camera with respect to some other, usually inertial, frame, as show in Figure II.2. This allows localization of objects in the scene using inertial frame, which is needed in order to be able to express the observer's position relative to the object in this inertial frame.

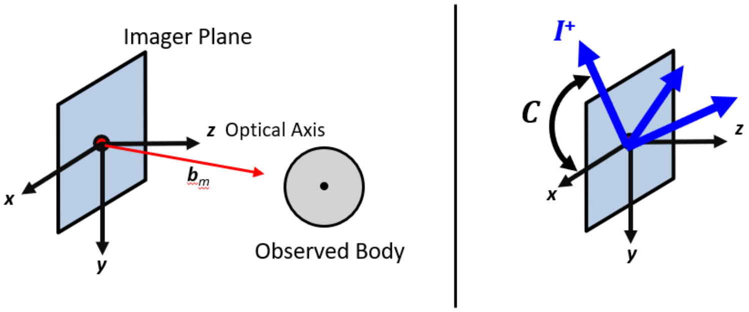


Figure II.2: Objects observed in a camera reference frame can be expressed in an inertial reference frame using a rotation matrix when the attitude of the camera is known.

Stars have been tracked and cataloged for thousands of years, and and tens if not hundreds of thousands are bright enough to be useful for attitude estimation for spacecraft. The reason why they are so useful for attitude estimation is that if you have a non-rotating frame, the apparent directions towards stars doesn't change as you move around because they are so far away. A spacecraft would need to move out to over 100 AU, past the orbit of Jupiter, for there to be any significantly noticeable parallax, or change in direction, towards stars. And even then, it's only the closest stars that will exhibit any change. The stars themselves are also moving, however again, the vast interstellar distances makes this movement barely apparent. The movement is also known and predictable, so catalogs can be updated as necessary. What all of this means is that, if an observer can look up at the night sky and identify what set of stars or constellation they are looking at, then they known their attitude in a non-rotating frame. Non-rotating frames include the Earth-Centered-Inertial (ECI) and the International Celestial Reference Frame (ICRF). Star catalogs, such as the Hipparcos star catalog, which is currently the most up to date and accurate catalog, use the ICRF frame to describe the directions towards the stars. So, when an observers identifies stars using Hipparcos, they known their orientation, or at least the orientation of their camera, in the ICRF. In reality, there are other factors and sources of error which impact the observed direction towards the stars. These include starlight aberration due to a fast-moving observer, gravitational lensing and distortion due to the presence of massive bodies near the observer of a star's light path to the observer, and a plethora of other hardware-related sources of error. Most of these fall outside the scope of this work and will not be discussed further.

One of the first challenges faced by the Orion OpNav system, was that of attitude estimation. Not because the algorithms, hardware, and methods needed for it aren't known, but because of difficulties in simultaneously observing the Moon and stars in a single image. The Moon is very bright compared to the surrounding stars, and previously the cameras used by spacecraft did not have the dynamic range to be able to observe dim stars and the Moon in a single frame without significant pixel saturation and blinding of the camera. There are several ways to work around this problem, one is to attempt to use high dynamic range (HDR) photography, another is to have a separate dedicated camera for observing the stars which is facing in a different direction than the Moon camera. The latter solution works, as long as the relative angle between both cameras is known, something which can be achieved if both cameras are mounted to a single rigid frame or block of material, as is done with sensing and navigational equipment in the GNC (Guidance, Navigation, and Control) Bay of a spacecraft. It was assumed that this latter solution would be implemented on Orion, and so the question of attitude determination was removed from the development scope of the position estimation algorithms. This means that, in order to perform Moon-based Op-Nav, it would only be necessary to develop algorithms which accurately estimate the distance to the Moon, based on how large it appears, and accurately determine the direction towards the Moon in a camera reference frame. Phrased another way, it was only necessary to develop accurate ways of estimating apparent size of the Moon and accurate methods of centroiding.

Thus, the basic process of estimating absolute position of Orion using Moon-based OpNav can be summarized as

1. Identify the lunar limb pixels
2. Estimate apparent size and centroid location in captured image
3. Use size estimate and centroid location to calculate Orion's position relative to the Moon expressed in the OpNav camera reference frame
4. Using attitude knowledge from the stars and a separate camera system express Orion's position relative to the Moon in an inertial frame, either ICRF or ECI.
5. Using the time and date of the observation and Moon ephemeris data, translate

Orion's position estimate from being relative to the Moon to being relative to the Earth

The main thrust of research which will be discussed in the chapter is dedicated to the first two items, namely estimating the apparent size and centroid of the Moon and using that to calculate a Moon-relative position estimate in the camera frame. The next two sections detail the development of the methods which do this.

Assumptions Behind OpNav in this Dissertation

There are several assumptions which must be clarified regarding the scope and context of the OpNav problem.

Observed Bodies Are Known *A Priori*: All observed target bodies have been previously observed by astronomers and engineers and have known size and shape. For asymmetric bodies, their orientation is known as a function of time. All observed bodies can accurately be modeled as an ellipsoid.

Optics Calibrated and FoV is Appropriate for Distance: It is assumed that the optics have been calibrated and that issues due to lens distortion or projection can be resolved. Also, distance between the observing spacecraft and the target body is such that the body appears as an extended ellipsoid. Bodies which appear too small or too large and extend past the camera field of view require a different kind of processing in order to use optical navigation. Camera pointing is sufficiently accurate to capture the target body in the field of view, though clipping can occur. The optics are also appropriate for the task in that the selected focal length for the optics and pixel pitch (size) of the sensor allow for the observation of an extended body. As an example, an illustration of how large the Moon appears as a function of pixel pitch and focal length when imaged from the Earth is shown in Figure II.3.

All Necessary Frames ar Known: All coordinate frames relevant to the problem of navigation are known. This includes frames attached to the target body, as well as inertial frames, such as the ICRF and ECI frames. Rotational and translational transformations and matrices are known at all times.

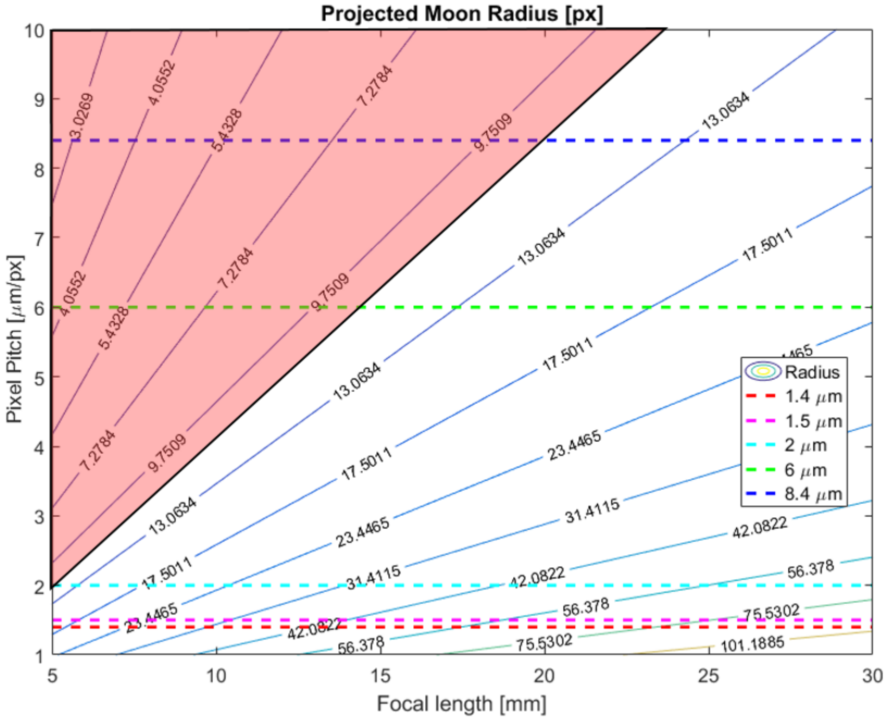


Figure II.3: Apparent radius of Moon in pixels as it appears from Earth fo different combinations of pixel pitch and focal length. Common pixel pitch values of commercial sensors are also shown. The area highlighted in red indicates configurations where the Moon appears too small to effectively use the proposed OpNav methods.

Background on Transition Across Object Edges in Digital Images

The nature of digital imaging is such that hard edges of objects, even those in focus, are described by a transition which covers a finite number of pixel on an imaging sensor. Instead of trying to define an infinitely thin line or curve which describes an edge, it is assumed that the edge is described by a transition in graytone (or color for RGB/CMYK images). This transition can be seen in images of the Moon like in Figure II.4.

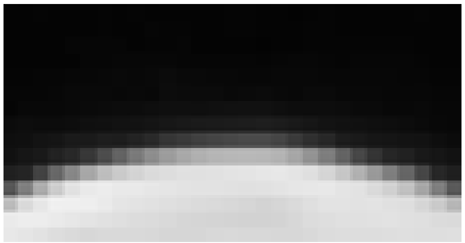


Figure II.4: Gradual transition in graytone across limb edge discretized by pixels.

If looking at a bright object against a dark background in a graytone image, such as the Moon against the background of space, the transition across the edge of the object can be described as a transition from a relatively high graytone to a relatively low one and there is some threshold value at which the exact edge can be defined. Below that threshold value, the pixel represent empty space, and above that value, the pixels belong to the object. However, the exact value of that threshold can be difficult to define. As seen in Figure II.5, even if the threshold is defined, the discrete pixel graytone will likely never take on the threshold value exactly.

If a continuous function is be used to describe the edge transition, then the exact

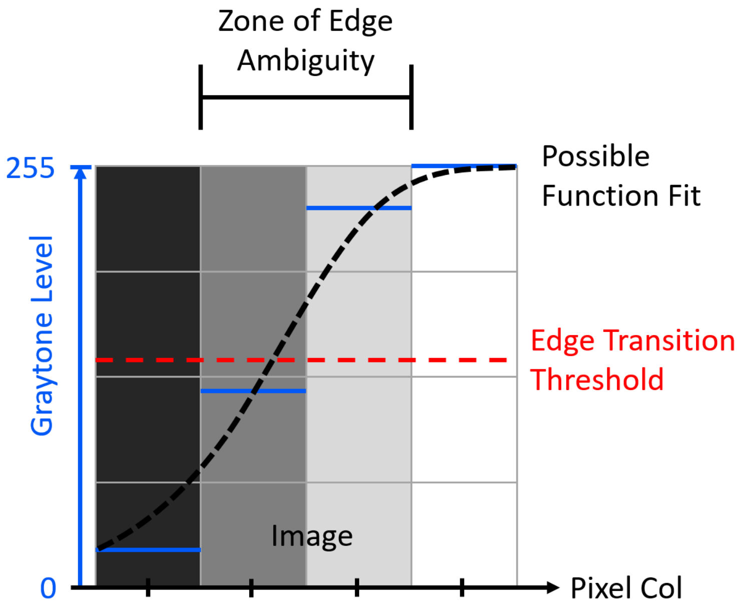


Figure II.5: Location of edge is ambiguous in digital images since no pixel values take on the transition threshold value. Blue lines in each column indicate graytone value.

point where the transition threshold is crossed can be located, allowing for more accurate edge locating. This increase in accuracy is the main innovation which was developed for Orion's OpNav system. However, there are two potential counterpoints to using such a scheme, both of which will be addressed. The first issue is that it is not clear how the exact level of the edge transition threshold should be decided. It's not clear if it should be the same for every edge, and there might be a dependence on scene lighting, the geometry, the material properties of object being observed, the optical properties of the camera. The second potential issue is that the choice of fitting function and the choice of how heavily to weigh graytone values during fitting can vary and it's not clear that there is one unique function which optimally models the transition across a body's edge in all situations.

The first concern is addressed by the fact that the goal of using such a function is to produce a consistent and precise fit for all the pixels which belong to an edge. That fit may not be accurate, but if the inaccuracy is the same for all edge pixels, and the entire edge is shifted consistently, then a simple calibration process using truth data can be used to correct such bias. The second concern is addressed by the fact that the scope of application of such a fitting function within this research is limited. It is being used only on the limbs of the Moon and the Earth. While the Earth and Moon do have different limbs - the Earth's limb has a softer transition due to the presence of an atmosphere - the lighting conditions at the brightly illuminated edge of the limb are consistent. The distance to the light source, the Sun, does not drastically change, both the Earth and Moon have rotational symmetry so lighting doesn't change as the bodies rotate, and the color and intensity of the light coming from the Sun also doesn't change. These constants lead to very consistent conditions for observation, thus it is appropriate to model the entirety of the body edge using a single function.

Setting the Stage – State of the Art, Literature Review, and Available Frameworks

This section is dedicated to providing the background and justification behind the methods initially implemented when the OpNav work for Orion began. At the time when the question of developing an autonomous backup navigation system for Orion was posed there were several theoretical frameworks that could be tapped for building the necessary navigation algorithms. State of the art navigational methods at the time had yet to take advantage of the onset of cheap high quality imaging sensors that had entered the market over the past several years. This left the door open for a wide array of different approaches of performing optical navigation.

State of the Art and Current Developments in OpNav

The motivation behind the development of OpNav for Orion really came from new requirements defined by NASA. Since the retirement of the Shuttle Program, NASA had stipulated that any new crewed missions must have an autonomous backup navigation solution for safety. The need for accurate position estimation was especially keen for Orion's missions around the Moon, where accurate position estimation is paramount when operating near the Moon and when heading back towards Earth. Entering the correct flight path corridor requires very accurate position estimation, which is difficult to do on-board. Following this incentive, the development work for Orion OpNav began in 2012. At the time, autonomous attitude estimation methods existed and had flight heritage, however systems for autonomous position estimation were lacking. There were some existing methods that relied on the imaging of distant planets to get a rough position fix, however these were not sufficiently accurate. These positioning techniques relied on estimating position relative to planets in our solar system which even under the best case scenario, would be at least 50 million kilometers away from any observer, which introduced large uncertainty into position estimates. This was unacceptable for Orion's navigational needs. Thus, a new technique was needed and so JSC started looking for new methods for performing position estimation in cis-Lunar space.

Dr. Mortari already had an existing relationship with some of the engineers at JSC which were part of the Orion development effort and they knew of his extensive work with star tracker algorithms as well as his existing work with the Moon-Sun sensor. The Moon-Sun sensor was originally proposed by Dr. Mortari in 1996. The goal of the project had been to use illumination of the Moon to estimate a direction towards the Sun in order to help get an attitude fix. The method had not been very accurate for producing attitude estimates however, much effort was dedicated to analyzing the illumination of the Moon, which would be necessary and come in useful when trying to use the Moon for position estimation.

Even though digital imaging systems have become a ubiquitous part of everyday life, first through dedicated digital cameras, and then through smartphones, they have not seen application in crewed space navigation beyond Low Earth Orbit. Digital cameras based on Charged Coupled Device (CCD) technology and Complementary metal-oxide-semiconductor (CMOS) sensors have become the standard for imaging needs. These technologies have been developed and miniaturized throughout the 1990s and 2000s and provide a readily available selection of commercial-off-theshelf (COTS) sensors which can be used for navigation needs of spacecraft.

When considered the total scope of all flown missions to space the application of optical navigation methods has been fairly limited, however, recent flagship missions such as NASA's Mars Reconnaissance Orbiter (MRO) and New Horizons as well as JAXA's Hayabusa have successfully implemented OpNav. Although OpNav was not mission critical for MRO, its optical navigation camera was successfully tested by imaging Mars' moons Phobos and Deimos [7]. New Horizon used techniques from Optical Navigation to refine estimates of the orbital parameters of some of the Jovian Moons [8] as it flew past them and then later again implemented OpNav near Pluto [9] . Finally, Hayabusa relied heavily on optical imaging for navigation as it neared the target asteroid, Itokawa [10].

Literature Review

An in depth review of existing optical navigation methods is provided by Owen [11]. This review makes it possible to see how the optical navigation methods being tested on Orion compare to existing navigation methods. At a minimum, optical navigation systems for spacecraft require knowledge of inertial position of some observed target as well as relative bearing and distance to the observer.

Light Time Effects: For the most general case, one must consider the effects of light time and stellar aberration. Standard formulations for how to take these effects into account are provided by the Explanatory Supplement to the Astronomical Almanac [12] These effects need only be included when considering a very distant target, therefore they are not used in the current version of the Orion opnav algorithms. The true position compensating for light time correction may be estimated from an iterative equation:

$$\mathbf{T}(t) = \mathbf{R}(t) - \mathbf{S}(t - \tau)$$

Where τ is the light time, $\mathbf{T}$ is the true position, $\mathbf{R}$ is the current position estimate and $\mathbf{S}$ is the position of the target. To correct for the stellar aberration, we use the following equation:

$$\mathbf{A}(t) = \mathbf{T}(t) + |\mathbf{T}(t)|[\dot{\mathbf{R}}(t)/c]$$

Where $\mathbf{A}$ is the apparent position and $\dot{\mathbf{R}}$ is the observer velocity in an inertial frame. This is a Newtonian formulation. If one is imaging stars and utilizing star catalogs, light time corrections for an observer located at the Solar System Barycenter (SSB) are already included in the coordinates of the star, thus simplifying position estimates.

Image Processing: Orion software follows a well established method of defining the camera frame. The same construction is offered by Owen. The camera attitude may routinely be described by a series of Euler rotation. Code to correct for cubic radial distortion due to the Gnomonic projection is available, however is not implemented on Orion. Standard transformations are used to convert from camera coordinates to imager coordinates.

Raw images may be corrected to remove pattern noise, pixel-to-pixel variations and dark current noise. A series of master bias images are taken that have zero second exposure times. A flat image is then taken and combined with the bias image. If the camera has thermal noise, additional dark images are combined to get a *master dark* image. Background can be removed using filtering that expects a certain type of image to be captured by the camera. For the purposes of the optical navigation, most images will contain mostly dark sky. Also, multiple images may be averaged together to remove hot pixels. Several different methods exist for performing centerfiding, or centroiding of observed targets. The method depends on the apparent size and nature of the target.

Centerfinding by correlation: Another method used in OpNav is centerfinding by correlation. This method works by defining a predicted brightness pattern as an array **P** and searching for that pattern within an observed array **D**. **P** may represent a point spread function (PSF) of a star, or a patch of terrain, or be detailed image data of an illuminated celestial body. The location of the **P** array is varied, and the location which produces the best correlation is the nominal image location. Subpixel accuracy can be achieved by using interpolation. This technique is called spatial correlation, while another method is frequency correlation.

Centerfinding by analytic function fitting: The centerfinding process can be done by performing a least squares fit of an analytic function on the image. The fitting function is nonlinear, therefore the process is iterative. A two dimensional guassian may be used. It has a defined peak location and standard diviation. The Lorentzian function is another frequently used point-spread function.

Limb Scanning: Used for targets that span more than a few pixels. The limb of a body is defined by the points which form the boundary of the body in the image. The vector connecting the observer to points on the limb will be tangent to the surface of the body. By selecting two points on the limb one can solve the vector equations of the triaxial ellipsoid and estimate the observer position.

Autonomy and software of NASA's Deep Space One: Bernard et al. offers a summary of the development and implementation of autonomous navigation on the Deep Space One (DS1) Spacecraft [13]. DS1, launched in 1998 served as a testbed for a multitude of new hardware and software systems. It was the one of the first implementation of a fully autonomous navigation system which, in addition, had code generated from MATLAB's stateflow toolbox and used autocoding to generate flight code. Also, verified with the help of module testing software based on automata theory that explored all scenarios that the code might fall in, including ones which developers had missed in testing. Behavior auditors were also used in the development. Anomaly detection that uses neural networks was also used. The image processing was anticipated to have a precision of 0.1 pixels although this was not initially achieved due to light scatter and problems with on-board instruments.

Raymon et. al summarize the capabilities and mission plan of DS1 [14]. DS1 used an on board ephemerids catalog of beacon asteroid, planets, and 250,000 stars from the Tycho catalog. Position estimation was provided by asteroid imaging which was used to compute a Heliocentric trajectory. Initial error was < 1000 km and 0.4 m/s. After image clean-up the 1σ error was 3 km. The imager used was the multi purpose Miniature Integrated Camera Spectrometer (MICAS) system which was used for scientific imaging as well as navigation. It collected light in a wide range of bandwidth, however was blinded in the ultraviolet due to noise from light reflecting off of MLI on the spacecraft. MICAS was tuned in flight and some parts of the code we even developed after launch due to tight scheduling.

Raymon et. al reviews the successful results from the DS1 mission. [15]. Despite multiple setbacks, and loss of the Stellar Reference Unit (SRU), autonomous optical navigation was adapted and allowed for the successful completion of all of the mission targets. The loss in the SRU did however mean that image processing on approach to Borelley had to be done on the ground, so on-board optical navigation was not performed. MICAS was used to replace the SRU for star imaging and target body tracking. It relied in stars appearing in the field of view when imaging the comet Borrelly. Initially imaging of the comet required co-addition of the images, though later on the comet became detectable in single images. Optical navigation ephemeris differed by 1500 km from the ephemeris computed at ground stations. Error in ephemeris was corrected when the brightest pixel was used to estimate the comet's position, instead of a Gaussian fit to the coma. The navigation team also faced difficulty getting a lock on the "Earthstar" used for navigation purposes. Prior to encounter, multiple sequences of photos were taking spanning a range of different integration times, since it was not known ahead of time what photometric properties of the comet were. The spacecraft did not have the capability to autonomously vary the integration time. At the encounter with the comet the autotracking system had difficulties keeping the nucleus in the FOV. This was due to a lag in ACS whenever the target exhibited significant angular acceleration. Tracking software found the nucleus in all except for 1 of the approach images. MICAS also had an infrared channel it used when imaging the comet. Lessons learned from the development of autonav for DS1 were applied to the Deep Impact mission.

Theoretical and Conceptual Frameworks

A large theoretical framework in optical navigation methods exists from which a number of solution approaches could be pulled for the purposes of Orion OpNav. A brief review of some image processing techniques, function fitting methods, and edge detection methods follows.

Image Based Position Estimation: Pose estimation using images of known markers is a classical technique. The problem is well known and understood and can be presented as follows: “Given a set of observed markers or features with known geometry and position, what is my (the observer) position and orientation with respect to the features?” When imaging geometries that can be described by a finite series of points represented in 3d space, the problem is known as the Perspective-n-Point (PnP) problem [16]. The question of being able to identify the individual points in the geometry is a whole other matter, and feature recognition is an entire field in of itself, but the crux of the PnP problem is 1) having enough points to be able to accurately estimate observer pose and 2) resolving some of the ambiguities which arise from projecting a set of points located in 3-D space onto a 2-D image. This is an active field of research and innovations in algorithms which solve the PnP problems are a present subject of investigation.

If the points, markers, or features which are observed and identified belong to a single rigid body of known size and geometry, then it is possible to estimate the pose of the observer with respect to that body at the time the observation was made. If the orientation, size, and position of that body is known with respect to the origin of some specified frame, then it is possible to estimate the pose of the observer with respect that origin. At long distances it may be difficult to resolve the orientation of an observed body, in which case the best an observer can do is to estimate their distance to the observed body based on how large (or small) it appears.

OpNav relies on this concept in order to estimate the position of Orion with respect to an Earth-centered frame. When operating near the Moon, Orion would be taking images of the Moon, and so OpNav techniques would provide the pose of the observer with respect to the Moon. However, the Moon is a familiar celestial object which has been tracked for millennia, and there is precise ephemeris data which tracks the exact position and orientation of the Moon with respect to Earth. So, if an observer can accurately estimate their position with respect to the Moon, they can also accurately estimate their position with respect to the Earth, which is helpful when crew are going around the Moon on their way toward Earth.

Numerical Methods and Least Squares Fitting for Attitude Estimation: Both linear and iterative non-linear least squares have applications in image processing and attitude estimation [17]. Almost all modern imagers use digital sensors, as opposed to celluloid film or other analog imaging methods, and thus edges of objects within a picture frame are described by a discretized transition of color or graytone. This discretization introduces error. By fitting a function to features on the image, objects in the frame can be described with greater precision.

One concrete example comes from star imaging and centroiding techniques. If a space-based optical system is well calibrated and focused to infinity when imaging stars, dim stars will only light up a single pixel. Thus, the center of that pixel could be defined as being the location of the star on the screen. This approach is inaccurate, however since discretization error due to having a digital sensor is introduced. To circumvent this issue, images are intentionally defocused so that light coming from stars is blurred and falls on multiple pixels. Afterwards, a Gaussian (or other desired function) can be used to describe the light intensity in the pixels as one moves away from the apparent star center. Once the peak of the Gaussian is located by using least squares, then the location of the star can be established to sub-pixel precision.

Edge and Shape Centroiding and Detection Algorithms: Many algorithms exist which are designed to look for certain features within a frame. Classical techniques focus on edge detection, corner detection, and the identification of simple geometries. Examples of such filters include the Canny Edge detector [18], [19], the Harris Corner Detector [20], [21], and Circular (or linear) Hough Transforms [22], [23]. There are extensive libraries and toolkits, such as OpenCV [24], with many other image processing tools dedicated to segmentation, centroiding, feature matching, and outlier rejection.

In recent times modern image classification techniques built from Convolutional Neural Networks (CNN), Deep Neural Networks (DNN), and Recursive Neural Networks (RNN) have exploded in popularity and application and are quite good at classifying contents of images. However, these methods are not suitable for highly precise, pixel-by-pixel level analysis of images and, perhaps even more importantly, lack the heritage and reliability of classical techniques to be used in high-risk spacecraft operations environments. Therefore, they were omitted when considering candidate algorithms for Orion OpNav.

Optics, Camera Frames, and techniques from Photography and Photo Editing: Fundamental concepts regarding camera optics, transformation of frames, the mathematics of projections, and photographic editing techniques are needed in order to be able to effectively make use of images for OpNav purposes.

Image editing techniques ranging from simple cropping, gradient filtering, and gamma correction to more complex noise reduction and dead pixel filters have application in this problem. Image preprocessing is an important step of preparing images for processing by reducing noise, resizing, rescaling of pixel levels, merging of channels, and perhaps even the removal or countering of filters which may have been performed automatically by camera software to make images appear more pleasing to the human eye but hamper image processing efforts.

Camera calibration and characterization is necessary in order to perform accurate analysis of images captured for OpNav. Pixel size, focal length, and field of view must all be known precisely. Also, any distortion due to the optical system must be characterized and understood through camera calibration procedures. Deviations from the optimal pinhole model camera, shown in Figure II.6, can be corrected for once proper camera calibration is complete.

Frame manipulation and projection describes the lossless transformation of image information from one frame to another as it passes through an optical system. In order to be able to accurately interpret a scene, it must be understood how light is transformed and projected from a 3-D scene onto a 2-D plane. Gnomonic projection effects [25], such as vignetting, may need to be corrected for. Clear definition of

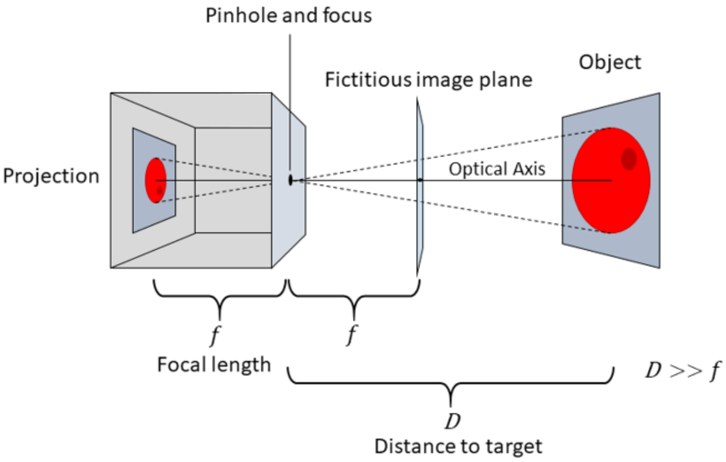


Figure II.6: Pinhole camera model used behind image processing algorithms.

the axes of the various image, sensor, camera, and body frames must be provided in order to correctly map real-world objects represented in a 3 dimensional coordinate frame into a 2 dimensional image frame discretized into pixel rows and columns. Different ways of representing frame orientation, such as directional cosine matrices, quaternions, and Euler axis and angle are useful as well.

Attitude Representation and Estimation: Methods for representing attitude are necessary for expressing vectors in the appropriate frame. Attitude representations include Euler axis and angle, directional cosine matrix, Rodrigues parameters, and quaternions [26], [27]. Attitude estimation methods such as ESOQ-2 [28], and $\boldsymbol{q}$-method [29] are useful in determining camera attitude given observations of reference objects, such as stars.

CHAPTER III

OPTICAL NAVIGATION USING THE MOON AND EARTH - DESIGN OF POSITION ESTIMATION ALGORITHMS AND FORMULATION OF FITTING FUNCTIONS

This chapter opens with a brief introduction of how the Orion OpNav algorithms are used to produce a position estimate. This is followed by a detailed description of each of the algorithmic components of the Orion OpNav system. Emphasis is placed on the implementation of the functions used to describe the Moon and Earth edge transition as well as the iterative non-least squares algorithms used to precisely fit those function onto the digitized edges of the Moon and Earth. Much of the foundational work behind these algorithms was performed by Mortari [30], [31] and Borissov [32], [33]. This work is expanded upon, elaborating further on certain aspects of image preprocessing as well as refining and simplifying aspects of the limb fitting models.

The overall image processing architecture is captured in a flowchart in Figure III.1 with a breakdown of each of the steps presented in the following sections. The first 5 steps of the image processing algorithm represent the initial processing and limb fitting. They are followed by the final pixel selection and the iterative nonlinear least squares.

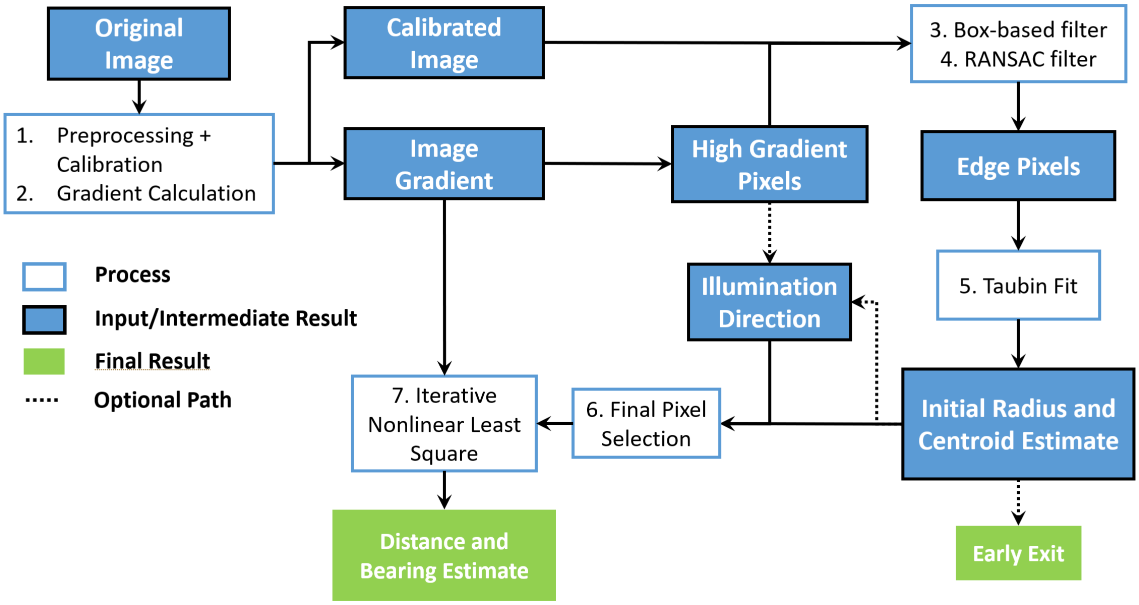


Figure III.1: Overview of OpNav Image Processing. Steps 1 - 5 correspond to generating the initial limb fit. Steps 6 and 7 are required for the final refined limb fit and position estimate.

Overview of Algorithm Steps 1 - 5 – Initial Pixel Selection and Limb Fitting

The Moon is well approximated by a sphere. It is more spherical than the Earth, which is not considered a sphere but an oblate spheroid with a flattening of about 0.34 percent. It can be shown that both of these bodies when projected onto a flat image plane through the gnomonic projection of a pinhole camera produce ellipses.

Therefore, the centroiding and sizing techniques used had to estimate both an apparent body center as well as semi minor and semi major axis in order to have maximal accuracy. In particular, for the Moon the necessity of measuring both semi minor and semi major axis is a function of how far the Moon appears off of the optical axis. If the Moon is centered on the optical axis of an image, then it appears as a circle and thus has a single radius measure. As it moves off axis, however, the projected ellipse becomes more oblate. This effect is particularly noticeable for cameras with a wide field of view with distortion becoming more and more pronounced as the Moon appears near the edges of the frame.

The image processing responsible for estimating apparent size and centroid were divided into two stages. The initial phase dealt with image cleanup, pixel selection, and initial fitting, while the second phase applied an iterative nonlinear least squares refinement to the first. Much of what is stated here can be found in the work by de Dilectis [34] in Chapter Two, however to be comprehensive and convenient it is covered in the following sections.

Assumptions Regarding the Scene

Before diving into the details of image processing algorithms, it is necessary to understand the assumptions made regarding the scene that would be processed.

Moon Alone in Scene with Dark Background: The Moon is the only bright extended object in the scene. This means that any other light sources in the image are point sources (stars), noise, or optical aberrations, such as lens flair. This assumption holds true for most scenarios of a spacecraft navigating cis-lunar space. There are cases where the camera might see other pieces of the spacecraft in the field of view. In fact, as is described in later sections, images of the Moon captured from the ISS were used during algorithm testing, and in these images portions of the space station are visible. Though it was not an explicitly stated requirement to be able to handle this sort of imagery, the algorithms did manage to pick out the Moon from the image, even though there were round pieces of the station that were visible that could have potentially confused the algorithms. The other possible scenario for seeing other bright objects other than the Moon would be when returning from the Moon and looking towards Earth. In this case it would be possible to see both the Earth and Moon, so a way to discriminate each would be necessary, especially in cases where one may be partially occluding the other. These cases, however were not tested, and are a subject of future work.

Grayscale Processing: All processing can be done in gray scale. RGB data is not necessary for image processing, however it is possible that additional information could be collected if each color channel is analyzed separately. For example, it's possible that light aberration of the optical system could be mapped out by seeing how different color channels are affected differently. This, however, has not been tested. All algorithm descriptions and formulations are presented assuming grayscale images. However, there is no reason why the methods could not be extended to color images as well.

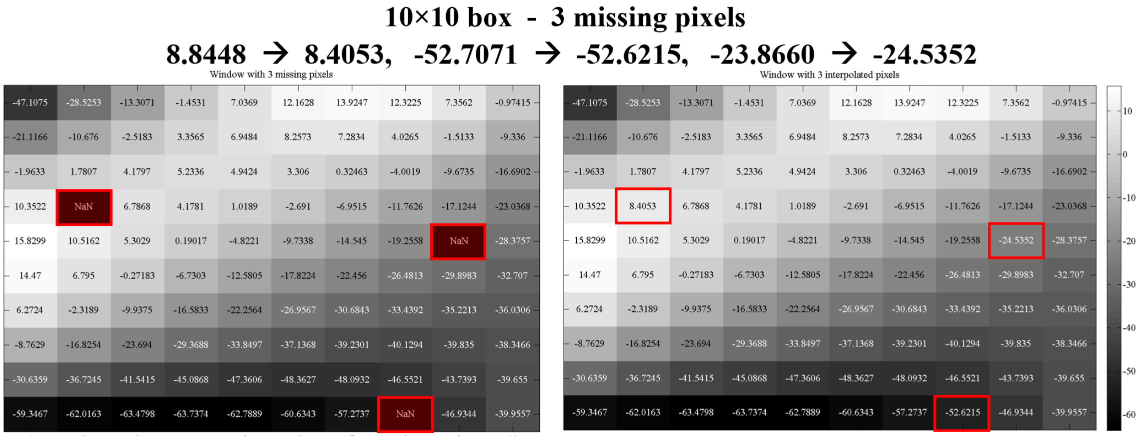
Camera Attitude and Time Known: Camera attitude and accurate time is known. This is provided by star trackers on board the spacecraft or through other external means. If a star tracker is providing the attitude for the camera, then it is assumed that it is mounted in a rigid way so that the orientation of the star tracker with respect to the OpNav camera is known and fixed. Accurate Moon, Sun, and planet ephemeris is available and up to date.

Surface Features Ignored: The surface features of the Moon are not of interest. It is sufficient to identify the limb in order to perform position estimation. Surface features could be used, however, in cases where the observer is very close to the Moon, or when camera attitude is not known and thus surface features, if correctly identified, could be used to help estimate observer orientation. This is outside the scope of this work.

Image Preprocessing – Dead Pixel Cleanup and Camera Calibration

After capturing an image of the Moon, the first step of image processing is to perform preprocessing which removes optical distortion and cleans up dead or saturated pixels. Dead and saturated pixels are pixels which inaccurately report the brightness of the incoming light. Dead pixel either report a zero value gray tone, or a constant low value while saturated pixels always report a maximal or constant bright gray tone value. These pixel usually stand out in sharp contrast to their neighboring pixels, however it may take a sequence of multiple images stacked together to make the these pixel more apparent. Once the faulty pixel have been identified local interpolation using Chebyshev polynomials is applied in order to "smooth" them over by using information from adjacent pixels [35]. An example of this interpolation used on dead pixels can be seen in Figure III.2

All image processing after this step assumes a pinhole camera model, so the image must represent a pinhole camera image as closely as possible. The optical distortion is handled first by calibrating the camera used to capture Moon images using calibration techniques such as those described by Zhang [36]. There are a



Window with 3 missing pixels

-47.1075	-28.5253	-13.3071	-1.4531	7.0369	12.1628	13.9247	12.3225	7.3562	-0.97415
-21.1166	-10.676	-2.5183	3.3565	6.9484	8.2573	7.2834	4.0265	-1.5133	-9.336
-1.9633	1.7807	4.1797	5.2336	4.9424	3.306	0.32463	-4.0019	-9.6735	-16.6902
10.3522	NaN	6.7868	4.1781	1.0189	-2.691	-6.9515	-11.7626	-17.1244	-23.0368
15.8299	10.5162	5.3029	0.19017	-4.8221	-9.7338	-14.545	-19.2558	NaN	-28.3757
14.47	6.795	-0.27183	-6.7303	-12.5805	-17.8224	-22.456	-26.4813	-29.8983	-32.707
6.2724	-2.3189	-9.9375	-16.5833	-22.2564	-26.9567	-30.6843	-33.4392	-35.2213	-36.0306
-8.7629	-16.8254	-23.694	-29.3688	-33.8497	-37.1368	-39.2301	-40.1294	-39.835	-38.3466
-30.6359	-36.7245	-41.5415	-45.0868	-47.3606	-48.3627	-48.0932	-46.5521	-43.7393	-39.655
-59.3467	-62.0163	-63.4798	-63.7374	-62.7889	-60.6343	-57.2737	NaN	-46.9344	-39.9557

Window with 3 interpolated pixels

-47.1075	-28.5253	-13.3071	-1.4531	7.0369	12.1628	13.9247	12.3225	7.3562	-0.97415
-21.1166	-10.676	-2.5183	3.3565	6.9484	8.2573	7.2834	4.0265	-1.5133	-9.336
-1.9633	1.7807	4.1797	5.2336	4.9424	3.306	0.32463	-4.0019	-9.6735	-16.6902
10.3522	8.4053	6.7868	4.1781	1.0189	-2.691	-6.9515	-11.7626	-17.1244	-23.0368
15.8299	10.5162	5.3029	0.19017	-4.8221	-9.7338	-14.545	-19.2558	-24.5352	-28.3757
14.47	6.795	-0.27183	-6.7303	-12.5805	-17.8224	-22.456	-26.4813	-29.8983	-32.707
6.2724	-2.3189	-9.9375	-16.5833	-22.2564	-26.9567	-30.6843	-33.4392	-35.2213	-36.0306
-8.7629	-16.8254	-23.694	-29.3688	-33.8497	-37.1368	-39.2301	-40.1294	-39.835	-38.3466
-30.6359	-36.7245	-41.5415	-45.0868	-47.3606	-48.3627	-48.0932	-46.5521	-43.7393	-39.655
-59.3467	-62.0163	-63.4798	-63.7374	-62.7889	-60.6343	-57.2737	-52.6215	-46.9344	-39.9557

Figure III.2: Interpolation of 3 dead pixels. Original values before removal: 8.8448, -52.7071, -23.8660. Values after interpolation: 8.4053, -52.6215, -24.5352

multitude of available tutorials and code for various proprietary and open source languages available online for performing this calibration. The calibration processes produces a camera calibration matrix which can be applied to the image to correct radial distortion including barrel and pincushion distortion as well as tangential distortion due to misalignment in the optics. The correction is performed by shifting and counter-distorting the pixel values in the image in order to produce and image which approximates an image that was captured by a pinhole camera. Note that such an image will still exhibit distortion due to gnomonic projection.

Image Gradient and Limb Pixel Selection

Following this preprocessing comes edge detection and selection of pixels belonging to the limb of the Moon. The goal of this process is to use these pixels to approximate a circle or ellipse. The size of the apparent circle or ellipse will be used for a distance estimate and can also seed a secondary, more refined, pixel selection process. This is done by first taking the gradient of the image, as shown in Figure III.3. There are many tools available for performing image gradients [37]. The chosen method was to take a 4 point central difference method using a 4x4 pixel kernel matrix and a single application of Richardson Extrapolation which approximates what the numerical image derivative would be if a smaller step sizes were used. This is a computational acceleration technique which provides a hi-fidelity image gradient at a relatively low computational cost.

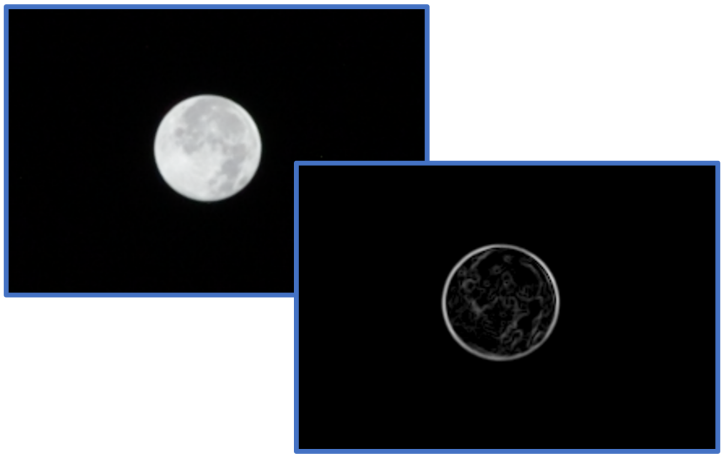


Figure III.3: Gradient filter applied to image of Moon.

As can be seen in Figure III.3, the brightest pixels on the image gradient tend to belong to the limb of the Moon. This may not always be the case, however, especially in cases where the Moon is only partially illuminated (most cases). Due to variations in surface topography there will be many high contrast pixels located around the terminator on the Moon where craters, peaks, and valleys will create areas of bright illumination adjacent to sharp dark shadows. Empirical tests are done to figure out the selection cutoff threshold for the brightness of the gradient pixels. However, simple thresholding is not enough to ensure that only limb pixels are selected. One approach to limiting the number of chosen pixels is to use an a-priori estimate of the expected size of the Moon in order to determine how many pixels should be selected. The number of selected pixels should be roughly half the circumference of the Moon, measured in pixels, since except for the case of a full moon, there is is normally only about a half-circle worth of limb visible. This prevents too many pixels from being selected, but doesn't help in the case where surface topography exhibits higher contrast than the limb. This problem is exacerbated when the same methods are used on Earth, as can be seen in Figure III.4. Clouds and continents create sharp boundaries all along the surface of the Earth while the atmosphere actually softens the limb edge, making thresholding more difficult and preventing limb pixels from being selected for processing. Additional filters are necessary in order to compensate for these issues.

Box Filter for Eliminating Outlier Candidate Pixels

A box filter is applied individually to every single selected candidate limb pixel in order to eliminate pixels which do not belong to the limb. This filter relies on a few assumptions regarding the pixels surrounding a potential limb pixel:

1. Limb pixels should not exist in isolation. There should be adjacent bright pixels in the image gradient which are part of the limb as well.
2. Assuming the candidate pixel is centered in the box, then there should be two opposing corners of the box which are both dark in the image gradient.

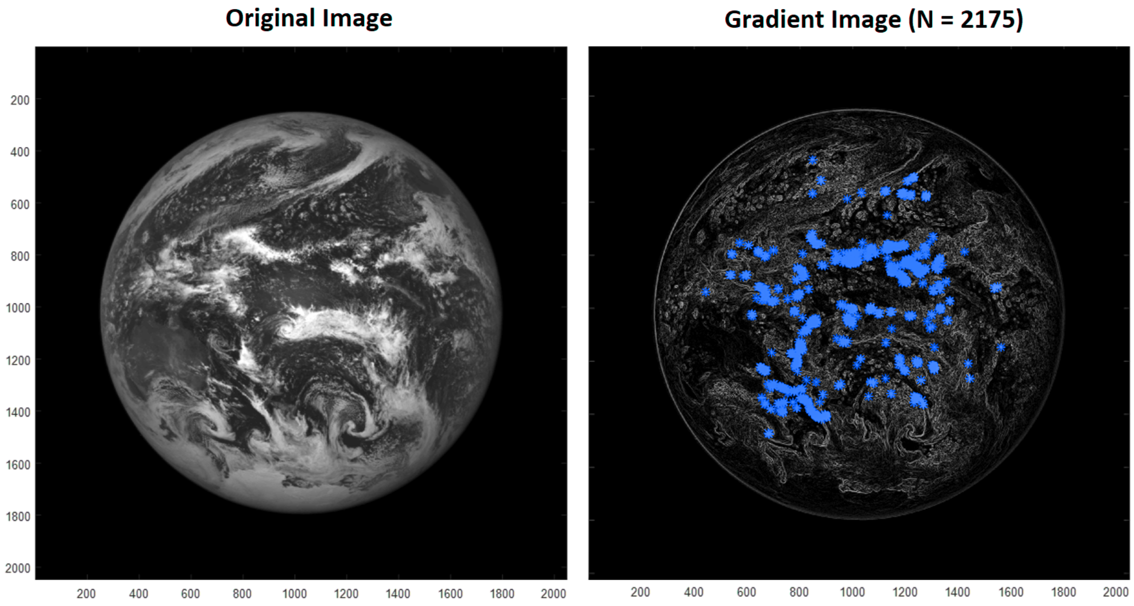


Figure III.4: High contrast pixels selected for image of Earth taken by DSCOVR. Note that none of the selected pixels in blue belong to the limb

3. The same two corners which were both dark in the image gradient should actually differ significantly in gray tone in the original image, as one should be in space while the other should belong to the illuminated portion of the body.

With these assumptions in mind a box ranging from 4x4 pixels up to 9x9 pixels is defined around a candidate pixel. The inertia tensor of the greytone values inside the box is computed and from that the principals axes are derived. One principal axis should be in line with the limb and where it intersects the box boundaries there should be bright gradient pixels. Once this first check is passed assumptions 2 and 3 are also tested by checking the corners farthest from the axis associated with the limb. If all three checks are passed, then the candidate pixel is kept, otherwise it is discarded. Figure III.5 shows a 7x7 box filter applied to a candidate pixel.

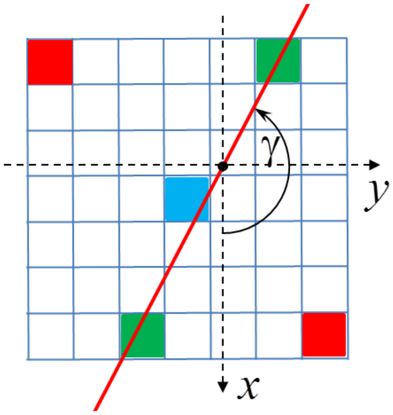


Figure III.5: 7x7 Box filter applied to a candidate pixel. Green pixels denote limb pixels while red pixels denote pixels farthest from the limb.

RANSAC for Eliminating Outlier Candidate Pixels

RANSAC, an abbreviation of "RANdom SAmple Consensus", is used next to eliminate candidate limb pixels that don't belong to a circle (or ellipse) [38]. RANSAC is an iterative fitting method which can be applied when trying fit noisy data that contains outliers to various mathematical models. RANSAC works by randomly sub-sampling groups of data from the available population and attempting to fit a mathematical model to that subset. If RANSAC detects a poorly fitting subset of the data, that data is labeled as an outlier and removed from the population and has no influence on the final fit. Once the RANSAC routine is ran on the candidate pixels, any pixels which don't fit the mathematical model of a circle or ellipse (according to a distance threshold) are removed. In order to accelerate RANSAC and produce more robust results, it's useful to initialize it with knowledge of focal length and an order-of-magnitude approximation of distance to the Moon which allows an observer to roughly guess how large the Moon should appear in the image. The final candidate pixels which survive the RANSAC filtering process are used for the Taubin best fitting which is initialized with estimates of centroid and radius of the apparent moon from the results of RANSAC.

Initial Taubin Least Squares Fit

Once the pixel selection is complete, and outliers not belonging to the lunar limb have been removed, an initial algebraic circle fit using the Taubin method is done. For initial fitting purposes, it is admissible to assume that the projection of the full Moon into the image plane is a circle. This would technically only be true if the Moon is centered in the field of view, however for the relatively narrow field of view of the images being processed and for the accuracy required of the initial fit, this simplification is appropriate.

The Taubin method uses a fast non-iterative algebraic least squares fitting in order to estimate the parameters *a, b, c, d* in the implicit equation of a circle: Eq. 3.1. It is robust in the sense that it works well even when only a small portion of the lunar limb is visible. Knowing the parameters of this equation also reveals the estimated centroid and radius of the observed Moon.

$$a(x^2+y^2)+bx+cy+d = x^2+y^2+\alpha x+\beta y+\gamma = 0 \tag{3.1}$$

Users have the option to exit with this initial fit result if it is sufficiently accurate/precise for their position estimation needs. If more precision is required, these results can be fed into other algorithms which refine the centroid and apparent size estimate further.

Step 6 – Estimating Sun Direction and Preparation for Nonlinear Least Squares

With the Taubin result in hand, the next step is to select a band of pixels for performing a refined fit of the lunar limb that should provide a more accurate estimate of the centroid and apparent size using iterative nonlinear least squares. These pixels capture the transition in graytone across the most strongly illuminated portion of limb to which we seek to fit a function.

The pixel selection process requires knowing or estimating the direction of solar illumination so that only the most strongly illuminated portion of the limb is considered for processing. This can be achieved in two different ways. First, it may be known *a priori*. If the attitude of the camera is known in an inertial frame and there is ephemeris for the Earth, Moon, and Sun, the it is easy to represent the vector connecting the center of the target body (Earth or Moon) to the center of the Sun in the camera body frame. Then that vector can be projected into the image plane. The second method of estimating the direction of illumination relies only on the existing results from image processing. The direction towards the Sun can be approximated by drawing a line which passes through two points, the first being the estimated target body centroid which was already estimated using the initial Taubin fit, and the second point being the centroid of the illuminated limb pixels which were used for creating that initial fit. This estimate may be off by several degrees from the true Sun direction, however it only needs to be accurate to within about 10 degrees for the purposes of establishing the selection band for nonlinear least squares. This is illustrated in Figure III.6.

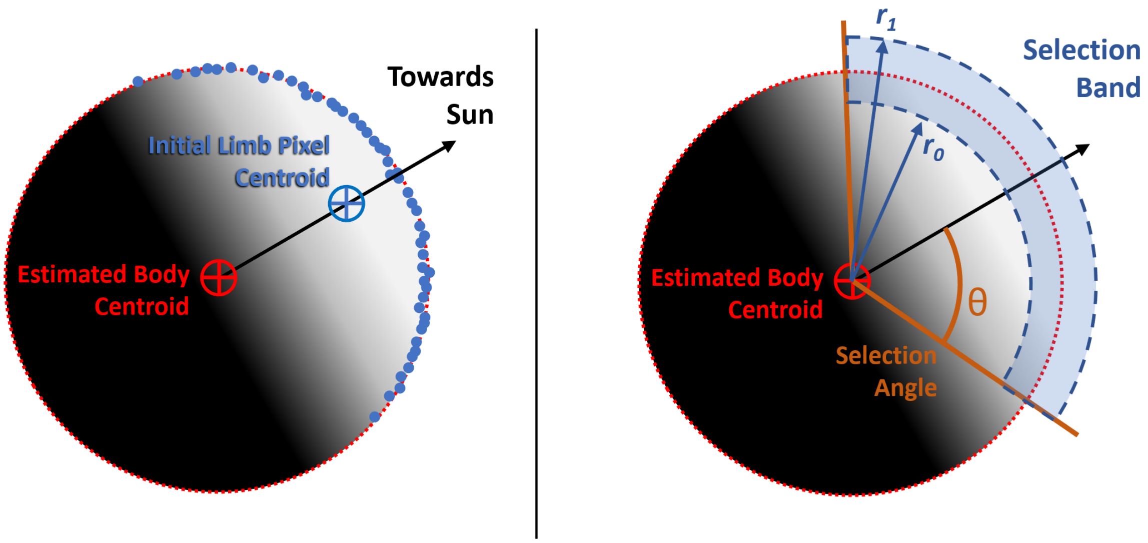


Figure III.6: Estimating Sun direction and selecting final pixel band. **Left:** Estimating the direction towards the Sun when ephemeris or attitude not available using estimated body centroid and initial limb pixel centroid **Right:** Defining a selection angle θ and radius range $[r_0, r_1]$ for final pixel selection band

A selection bounding angle of $\theta < 70°$ on either side of the sun direction vector is used to limit the size of the band to prevent selection of pixels belonging to lunar surface near the terminator which may be poorly illuminated. The width of the band is determined by an internal and external radius, $[r_0, r_1]$. This is measured by how many pixels on either side of the limb are desired given the initial radius estimate r. Generally, the closer and larger the target body appears, the wider the band of pixels necessary. For up close images with high resolution, this may be up to 10 or 11 pixels on either side of the initially estimated limb radius. For target bodies which appear small this may be limited to 3 or 4 pixels on either side of the initial radius estimate.

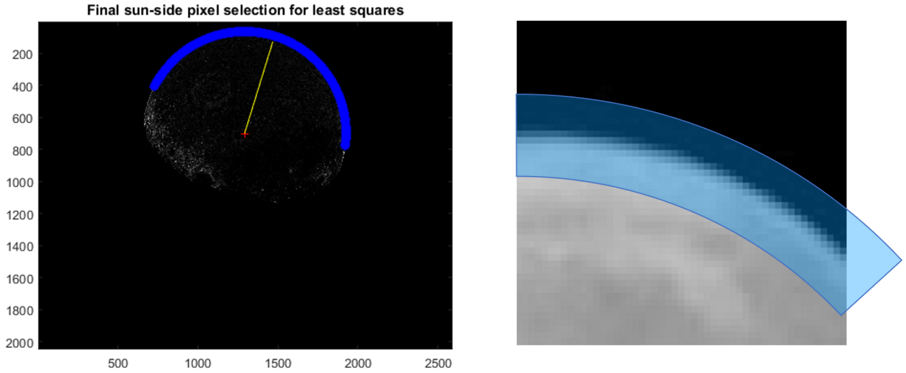


Figure III.7: Pixel selection on real image for nonlinear least squares. **Left:** Direction of solar illumination and selected limb pixels shown on image gradient. **Right:** Close up showing the selection band

Figure III.7 shows the same selection process described above but applied to a real Moon Image. The exact relationship between observer to target distance and the size of the selection band is empirically determined and would also depend on whether or not the target body has an atmosphere. For example, Earth's atmosphere makes the edge transition appear more gradual and thus a wider band of pixels is needed. Also, if one is imaging a particularly oblate body, or there is significant distortion due to gnomonic projection, the width of the band needs to be sufficient to account for the fact that the limb doesn't follow a circular arc. The band needs to be wide enough so that the limb does not get too close to the edge of the selection band. Once the limb pixels have been selected for nonlinear least squares, there are two different frameworks that can be implemented in using these limb pixels. The two frameworks are the 2D Image Plane Approach to Position estimation, the other is the 3D Triaxial Body Approach.

2D Sigmoid Nonlinear Least Squares for Position Estimation

The 2D Sigmoid approach fits a function to the flattened image of the observed body in order to produce a position estimate in the camera reference frame. This approach is suitable for bodies that are relatively smooth surfaces and not too oblate. The Moon is a suitable candidate, since for all intents and purposes it appears as a sphere at distances greater than 20,000 km. Its surface is not very smooth, however, and this will create some issues that will be discussed in a later section.

Mathematical Formulation of Limb Fitting and Graytone Transition Across 2D Scene

The formulations in this section have been described by Mortari [31]. These are re-derived and explained here and their shortcoming are addressed in the subsequent sections. Figure III.8 gives an idea of what the lunar limb looks when imaged with a narrow field of view lens from Earth. The image is relatively well in-focus so the transition across the edge take place over about two to three pixels. It is on images similar to this one that the nonlinear least squares would be applied.

To understand how nonlinear least squares can be used in the 2D image plane it is helpful to recall the equation for a sigmoid curve, specifically the logistic function, though there are many different sigmoid curves which all exhibit a similar “S” curve shape. The equation for the logistic function defined linearly along the x-axis is:

$$S(x) = \frac{1}{1 + e^{-x}}$$

This function describes a transition across the origin from one asymptotic level

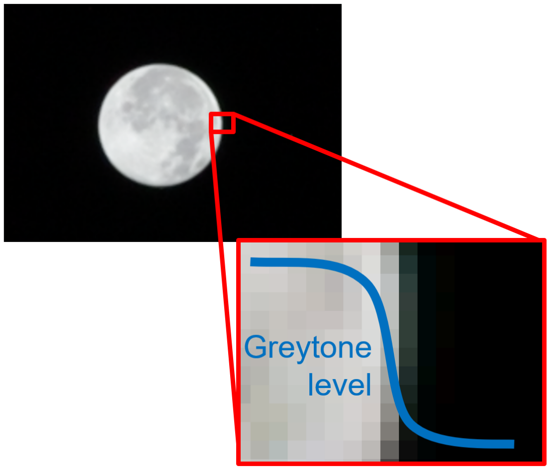


Figure III.8: Depiction of Sigmoid variation in pixel brightness as one transitions across the lunar limb into space.

to another (-1,1) or vice versa, (1,-1) depending on the sign in front of the x in the exponent. The function is easily scaled and shifted along the x-axis and y-axis so the function can describe a transition from any level to any other level at any x-coordinate over any length of transition section. This more generic formulation looks like:

$$S(x) = \frac{\alpha}{1 + e^{-k(x - x_0)}} + \beta$$

where α, β, k, and x_0 are all real numbers. The form shown above readily can be used to describe the transition in graytone across a vertical edge in a picture, where the normal of the edge is along the x-axis of the image frame. However, a modified formulation is necessary for any edges which aren't vertically aligned, such as the edge of a disk as shown Figure III.9

In a perfect disk, for any point on that disk, the direction towards the center

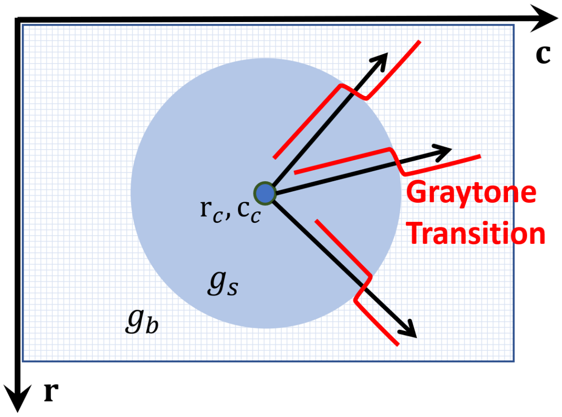


Figure III.9: Graytone transition along edge of a circle. r and c indicates pixel rows and columns. r_c and c_c indicate the circle center.

along which distance from the center is measured is also normal to a circle passing through that point. Put another way, the spokes of a wheel are all normal to the wheel. Now, assume there is a disk which exhibits a uniform graytone transition all around its circumference, meaning that the disk is uniform in color and brightness, the background is uniform as well, and the disk's edge has uniform sharpness. The distance from the center of the disk to the midpoint of the sigmoid transition will be the same around the circumference. For such a disk a circular formulation of the sigmoid function can be used, where the single x coordinate of the original formulation is replace by a coordinate pair $[r_i, c_i]$

$$S(r_i, c_i) = g_b + \frac{g_s - g_b}{1 + e^{k(r_e - \sqrt{(r_c - r_i)^2 + (c_c - c_i)^2})}} \tag{3.2}$$

where $\sqrt{(r_c - r_i)^2 + (c_c - c_i)^2}$ is the distance of a chosen pixel with coordinates $[r_i, c_i]$ from the center of the circle, r_e is the expected radius of the disk, g_b is the graytone of the background and g_s is the graytone of the surface of the disk. The formulation in Equation 3.2 describes a sigmoid transition with a midpoint located on the expected radius, r_e of the disk.

However, it is common when imaging planets to observe an ellipse. This can occur either when an ellipsoidal object is observed anywhere in the field of view, or when a spherical object appears somewhere far away from the optical center. The edge of the ellipse is a variable distance from the center, so a single function can not be used to describe the transition for all edge pixels if it only uses the distance to center as parameter. The distance to center must be transformed according to where along the perimeter a pixel lies. The previous form, Equation 3.2 can still be used if a linear transformation is applied to the edge coordinates to squish the ellipse into a sphere. The general equation for an ellipse with center $[r_{ec}, c_{ec}]$ and semi axis length $a,\ b$ is

$$\frac{(r_i - r_{ec})^2}{a^2} + \frac{(c_i - c_{ec})^2}{b^2} = 1$$

where b can be rewritten in the form

$$b = \rho a$$

which allows the equation of the ellipse to be rewritten in the form

$$(r_i - r_{ec})^2 + \frac{(c_i - c_{ec})^2}{\rho^2} = (r_i - r_{ec})^2 + (w_i - w_{ec})^2 = a^2 \tag{3.3}$$

where $w = c/\rho$. We will call this the transformed ellipse as this equation changes the equation of the ellipse to look like that of a circle, allowing the use of Equation

3.2 but with transformed coordinates w instead of c and semi major axis length a instead of radius r_e.

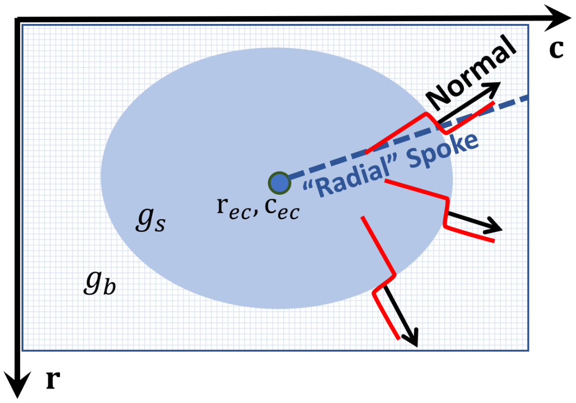


Figure III.10: Graytone transition along edge of an ellipse. “Radial Spokes” coming from the center aren’t normal to the ellipse at all points.

This transformation is still insufficient however to accurately describe the transition on either side of the edge of the ellipse. As can be seen in Figure III.10, the “radial spoke” coming from the center and intersecting an edge pixel is only normal to the ellipse at the semi major and semi minor nodes. Everywhere else, it impinges at an oblique angle, and thus using the distance from center to describe the transition will not produce the same steepness of transition for all edge pixels. Spokes near the semi minor and semi major nodes would have relatively fast transitions. While spokes far away from the semi major and minor nodes will have longer, more gradual transitions. Thus, distance from the edge must be calculated along a local normal, which must be calculated separately for every pixel as well. This will be forthcoming referred to as the ellipse normal problem. This problem will only have noticeable impact for highly eccentric ellipses.

The final difficulty added when trying to analyze the edge transition of an apparent ellipse is that the ellipse is generally not centered in the field of view, and is inclined with respect to the image axes, as shown in Figure III.11.

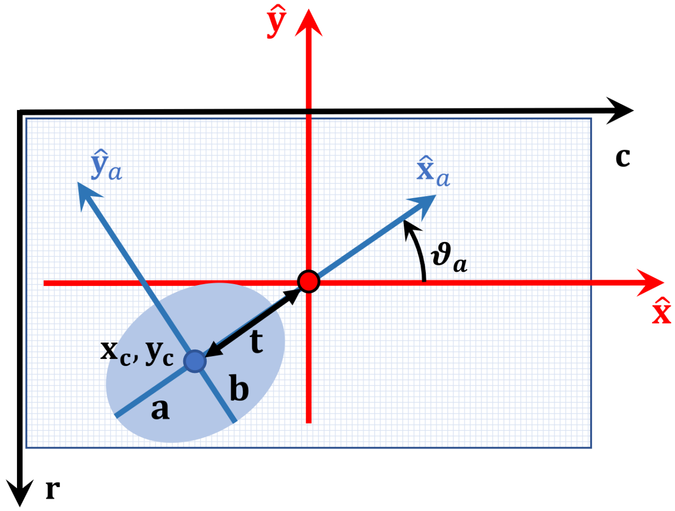


Figure III.11: Imaging an ellipse which appears rotated and off-center with respect to the image plane. A translation and rotation is necessary in order to describe a pixel location in semi axis coordinates. The ellipse center x_c, y_c is now defined in image plane coordinates instead of row-column coordinates

Thus there needs to be rotation θ in order to take a pixel described in image coordinates $[r_i, c_i]$ and describe it in the semi axis frame. For an $N_c \times N_r$ resolution image, this transformation is captured by the following equation

$$\begin{bmatrix} x_{ai} \\ y_{ai} \end{bmatrix} = \begin{bmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} c_i - (N_c+1)/2 \\ (N_r+1)/2 - r_i \end{bmatrix}$$

Where the transformation contained inside the right-hand vector is necessary to convert row-column pixel coordinate representation to x, y imager coordinates which have their origin in the optical center of the image, as shown in Figure III.11. Once this transformation is applied, Equations 3.2 and 3.3 can be used to describe the transition along the edge. Equation 3.3, however needs a minor adjustment before it is applied. We will define $w_{ai} = y_{ai}/\rho$ instead of the original $w = c/\rho$ formulation. Also, since we are now dealing with the transformed ellipse, a will appear in the exponential, instead of the previous r_e. With these fixes applied we have the elliptic sigmoid function:

$$S(x_{ai}, w_{ai}) = S_i = g_b + \frac{g_s - g_b}{1 + e^{k(a - \sqrt{(x_{ai}-x_c)^2 + (w_{ai}-w_c)^2})}} \tag{3.4}$$

This equations should be able to accurately describe the graytone transition over the edge of observed spherical body, while keeping in mind the aforementioned ellipse normal problem which would be an issue for bodies which appear highly ellipsoidal.

Cost Function Minimization and Iterative Update

We now prepare the sigmoid formulation for use with nonlinear least squares. Using Equation 3.4, and replacing k with κ to reduce future confusion with iteration counters, we can formulate a residual:

$$L_i = S_i - g_i = \left(g_b + \frac{g_s - g_b}{1 + e^{\kappa(a - \sqrt{(x_{ai}-x_c)^2 + (w_{ai}-w_c)^2})}} \right) - g_i \tag{3.5}$$

Peeking into Equation 3.5 it can be seen that minimization of L_i requires accurate estimation of the seven unknown parameters $\mathbf{u} = [g_b, g_s, \kappa, a, \rho, x_c, w_c]$ which means there are seven partial derivatives which must be calculated to populate the Jacobian for each selected edge pixel. An initial guess must also be provided for each parameter:

- g_{b0}: This represents the graytone of the surrounding background behind the body. It can be initialized as the average graytone of the pixels on the outer edge of the selection band.

- g_{s0}: This represents the graytone of the surface of the observed body. It can be initialized as the average graytone of all the pixels on the inner edge of the selection band.

- κ_0: This is the steepness of the transition across the edge of the body. For bodies that appear large in the field of view this transition may occur over several pixels, possibly up to ten, while bodies that appear small might only need about three pixels to describe the transition. If we assume, that on average roughly five pixels are sufficient to describe the transition, then an initial value of 1 is appropriate for κ. This can be empirically tuned.

- a_0: Is the estimate of the semi major axis of the observed ellipse and can be set equal to initial radius estimate produced by the Taubin fit.

- ρ_0: Most observed ellipses will not exhibit significant eccentricity, therefore it is sufficient to assume that ρ is initially 1.

- x_{c0}: This is initialized as the x-coordinate of the centroid estimate produced by Taubin in the initial fitting.

- w_{c0}: This is initialized as the y-coordinate of the centroid estimate produced by Taubin in the initial fitting. Since we are assuming an initial ρ of 1, there is no need to scale this coordinate.

L_i is a nonlinear function of the parameters $\mathbf{u}$. L_i is linearized and an iterative framework is set up using the initial values defined above.

$$L_i = L_i(\mathbf{u}) \quad \text{with} \quad \mathbf{u} = [g_b, g_s, \kappa, a, \rho, x_c, w_c]^{\mathrm{T}}$$

Linearizing L_i with respect to parameters $\mathbf{u}$ produces:

$$\begin{aligned} L_{i,k+1} =& L_{i,k} + \frac{\partial L_i}{\partial g_b}\Delta g_{b,k+1} + \frac{\partial L_i}{\partial g_s}\Delta g_{s,k+1} + \frac{\partial L_i}{\partial \kappa}\Delta \kappa_{k+1} \\ &+ \frac{\partial L_i}{\partial a}\Delta a_{k+1} + \frac{\partial L_i}{\partial \rho}\Delta \rho_{k+1} + \frac{\partial L_i}{\partial x_c}\Delta x_{c,k+1} + \frac{\partial L_i}{\partial w_c}\Delta w_{c,k+1} \end{aligned}$$

The linearized equation which must be solved in order to calculate update $\mathbf{\Delta u}$ is thus

$$\mathbf{0} = \mathbf{L}_k + \mathcal{J}_k \mathbf{\Delta u}_{k+1} \tag{3.6}$$

Where $\mathcal{J}$ is the Jacobian:

$$\mathcal{J} = \begin{bmatrix} \vdots \\ \dfrac{\partial L_i}{\partial \mathbf{u}} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & & \\ \dfrac{\partial L_i}{\partial g_b} & \dfrac{\partial L_i}{\partial g_s} & \dfrac{\partial L_i}{\partial \kappa} & \dfrac{\partial L_i}{\partial a} & \dfrac{\partial L_i}{\partial \rho} & \dfrac{\partial L_i}{\partial x_c} & \dfrac{\partial L_i}{\partial w_c} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

and the least squares update equation is:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{\Delta u}_{k+1} \quad \text{where} \quad \mathbf{\Delta u}_{k+1} = -\left(\mathcal{J}_k^{\mathrm{T}} \mathcal{J}_k\right)^{-1} \mathcal{J}_k^{\mathrm{T}} \mathbf{L}_k \tag{3.7}$$

The nonlinear least squares iteration is repeated until the stopping criteria

$$||\boldsymbol{\Delta}\mathbf{u}_{k+1}||_2 < \varepsilon$$

is reached, or a maximum number of iterations is reached. The value of ε must be empirically determined depending on user needs.

Position Estimate

From this refined estimate of the parameters $\mathbf{u}$ the ones that are relevant to position estimation are a, ρ, x_c, w_c where ρ is only necessary so that w_c can be converted to y_c. $[x_c, y_c]$ are the coordinates of the refined centroid estimate that are used to define a direction towards the observed body and a serves the same purpose of a radius estimate when observing a spherical body, though a more complex treatment is necessary if the body underwent significant gnomonic distortion. By comparing a, measured in pixels, and applying a pinhole camera model ... equation .... the distance to the observed object can be estimated (assuming its radius is known). Thus the position with respect to the body can be expressed in the camera frame. If the attitude of the camera is known, this position can be expressed in an inertial frame.

Limitations and Potential Modifications to the Method

This method faces several limitations. The immediate limitation is that it is not well suited for oblate or ellipsoidal bodies. Both spherical bodies and triaxial ellipsoids produce ellipses when projected into a 2-Dimensional surface [39]. However, unlike spherical bodies, the center of the projected ellipse one sees when imaging an oblate body is not necessarily in-line with the center of the body. Figure III.12 illustrates this concept.

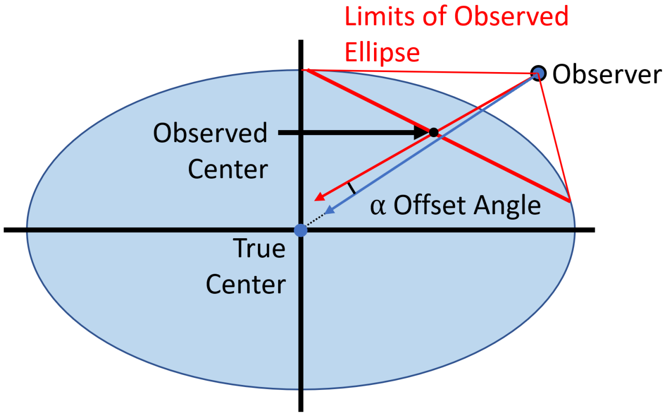


Figure III.12: Offset angle α between the apparent body center and true body center when imaging an ellipsoid.

If one is estimating their position with respect to the apparent center of an observed ellipsoid, vs the true center, then their position estimate will incur and error. The magnitude of that error is a function of distance to the body, latitude, and how oblate the body is. Attempts to find an analytic solution for offset angle α led to investigations in the geometric definition of triaxial ellipsoids in the hopes of reconciling the 2D sigmoid approach with oblate spheroids. Some possible corrections based on empirical results were proposed, however these corrections were never implemented.

Another closely related issue to the offset angle is the ellipse normal problem previously defined. Measuring distance from the center of the ellipse does not produce a path which is normal to the ellipse' perimeter and thus produces a skewed description of the edge transition. It is possible that this issue may be resolved using transformed ellipse coordinates, however this has not been verified.

Finally, for bodies which appear far from the optical center of the camera, it is inaccurate to only use a as an estimate for the apparent radius of the observed object. Instead, a must be scaled down appropriately to compensate for the gnomonic distortion which stretches out objects radially, and to a lesser degree tangentially. It may also be necessary to separately consider the semi major and semi-minor axes of the observed ellipse and might not be possible to accurately come up with a single parameter a which represents the size of the observed body.

Multiple investigations went into addressing some of these issues, however this method was instead wholly superseded by a more generic 3D formulation, presented in the next section, which can handle more oblate bodies and does not suffer from the difficulties introduced by gnomonic projection.

3D Sigmoid Nonlinear Least Squares for Position Estimation

The previous section developed a description of the transition across the edge of an ellipse using a two dimensional derivation, where all calculations are done one the projected image of an observed celestial body. That process was well suited to spherical bodies, however encountered difficulty when dealing with triaxial ellipsoids and also suffered losses in accuracy due to gnomonic projection. This section presents an entirely different formulation based on a three dimensional triaxial ellipsoidal body which also does not suffer from these issues. Unlike the two dimensional approach, the three dimensional approach produces a position estimate defined in the principal axes of the observed target body, not in the camera frame. This means that in order to use this method, the ellipsoid of the body must be known. The method would not work for a body with unknown semi-axis lengths. Also, if the goal is to provide a position estimate defined in some other frame, then the orientation of the axes of the observed body must also be known with respect to an inertial frame.

Geometric Derivation

This 3 dimensional formulation stemmed from attempts to address the limitations of the 2 dimensional approach, and eventually entirely superseded it. We begin with the definition of an observer position **p** defined with respect to a triaxial ellipsoid as shown in Figure III.13.

The equation for a triaxial ellipsoid is as follows:

$$\frac{x}{a^2} + \frac{y}{b^2} + \frac{z}{c^2} = 1$$

Where *a, b, c* are the semi-axes of the ellipsoid. This equations can be rewritten

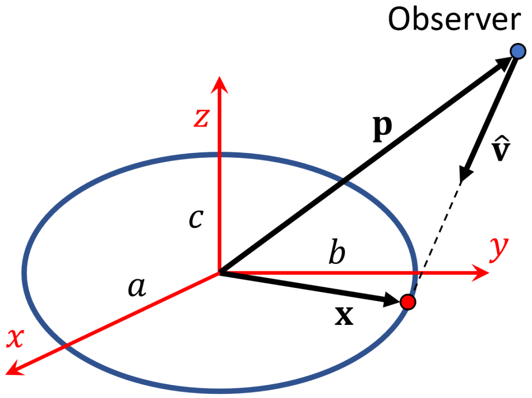


Figure III.13: Observer located at point $\mathbf{p}$ with respect to origin of triaxial ellipsoid. Vector $\mathbf{x}$ emanates from the center of the body and goes to a point on the surface. Vector $\hat{\mathbf{v}}$ point from the observer towards a point on the surface of the body.

in vector form

$$\mathbf{x}^{\mathrm{T}}\mathbf{J}\mathbf{x} = 1 \tag{3.8}$$

Where $\mathbf{x} = [x, y, z]^{\mathrm{T}}$ and $\mathbf{J} = \mathrm{diag}(a^{-2}, b^{-2}, c^{-2})$. The two formulations are equivalent in that they represent the same scalar equation, however the second formulation will be more useful moving forward.

When observing a triaxial ellipsoid, if one connects all the points along the observed limb to the observer, this will describe an elliptic cone. An observer picks one of these points along the limb and defines a vector pointing towards it as $\hat{\mathbf{v}}$. $\mathbf{x}$ can be defined with respect to $\mathbf{p}$ and $\hat{\mathbf{v}}$:

$$\mathbf{x} = \mathbf{p} + v\hat{\mathbf{v}} = \mathbf{p} + \mathbf{v} \tag{3.9}$$

The exact length v of $\mathbf{v}$ is not important just has to satisfy the constraint that $\mathbf{v}$ is tangential to the surface, meaning that it intersects the body of the ellipsoid only at one single point on the surface. In order to find the values of v that satisfy this constraint, one can solve Equation 3.8. The equation is quadratic in v and in order to satisfy the constraints, the two roots v_1 and v_2 must be equal. Substituting Equation 3.9 into Equation 3.8 and enforcing that the roots v_1 and v_2 are equal, one can arrive at Equation 3.10

$$\mathbf{v}^{\mathrm{T}}\mathbf{M}\mathbf{v} = \hat{\mathbf{v}}^{\mathrm{T}}\mathbf{M}\hat{\mathbf{v}} = 0 \quad \text{where} \quad \mathbf{M} = \mathbf{J}\mathbf{p}\mathbf{p}^{\mathrm{T}}\mathbf{J} - (\mathbf{p}^{\mathrm{T}}\mathbf{J}\mathbf{p} - 1)\mathbf{J} \tag{3.10}$$

This result is verified through eigenanalysis and is proven by Mortari [31]. Just like Equation 3.8, Equation 3.10 is a scalar equation. It is satisfied by any $\hat{\mathbf{v}}$ vector which represents a line-of-sight vector emanating from an observer towards the observed limb of a triaxial ellipsoid. If we relax that constraint and allow $\hat{\mathbf{v}}$ to point in towards the body or away towards space then we have the following Equation 3.11:

$$\xi = \hat{\mathbf{v}}^{\mathrm{T}}\mathbf{M}\hat{\mathbf{v}} \tag{3.11}$$

Equation 3.11 gives a useful method for measuring how far "off-limb" a certain vector $\hat{\mathbf{v}}$ may be. From an observer's point of view when $\hat{\mathbf{v}}$ points exactly towards the visible limb, Equation 3.11 is zero. When it points inward towards the body of an observed triaxial ellipsoid it becomes positive. When it points away from the body, ξ becomes negative.

The pixel selection process — Step 6 of the full OpNav algorithm — creates a band of pixels along the limb of an observed body. The center of each of these pixels, $[r_i, c_i]$ can be transformed into a corresponding $\hat{\mathbf{v}}_i$ vector which emanates from the focal point of the camera frame and points to the center of the pixel and this for every selected pixel p_i there is a corresponding graytone g_i and a corresponding

ξ_i. This sets us up to define a new functional description of the graytone transition across the edge.

Mathematical Formulation of Limb Fitting and Graytone Transition Across a 3D Scene

Previous discussions of graytone transition over an edge were based on the assumption that the transition occurs over a distance measured in pixel coordinates. We are moving away from this formulation and shifting towards looking at angular distance from an edge, as opposed to linear pixel distance. The same sigmoid function which describes a transition in graytone over a flat surface works just as well for a graytone transition over an angular sweep, and in fact, for small angles the two approaches are almost numerically identical. Deviation between the two approaches starts to arise when an object is significantly distorted due to gnomonic projection. The current formulation completely sidesteps this issue however. The only concern that might arise due to gnomonic projection is the the question of whether or not the band of pixels which sweeps along the limb is wide enough to capture a limb which might be stretched out due to gnomonic distortion. See Figure III.14 for more information.

When comparing two bands of equal pixel width, one at the optical center, the other far away from it, the two bands have the same width on the imager but subtend different angles. Even though the distant band might contain pixels which have been stretched out, this effect is reversed when the graytones are mapped onto angular displacement. Thus the distant band subtends a smaller angle and thus the

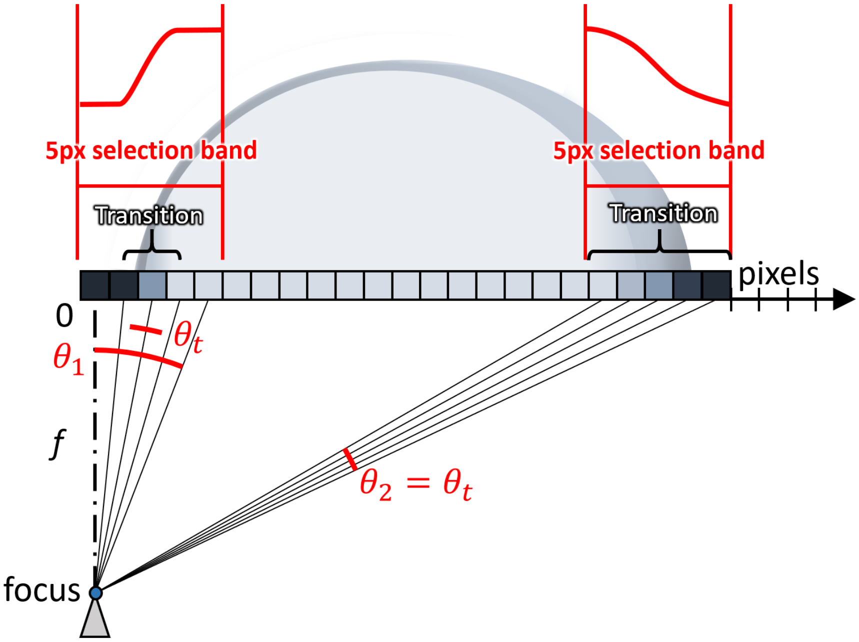


Figure III.14: Gnomonic distortion stretches out an edge transition which is far from the optical center. However, even though the transition spans more pixels, those pixels still subtend the same angular distance as a transition at the optical center of the image.

transition which has been stretched in the 2d image plane is now compressed back to its original width when transforming into angle-space. An analogous case would be that of a curved spherical imager. If pixels were placed on a spherical imager, instead of a flat one, then all edges of the same body would subtend the same number of pixels regardless of their location in the image. This is the effect being created by performing the analysis in three dimensions as opposed to two and this is the reason why it is worth formulating an angle-based sigmoid fit the the graytone transition across a limb.

Angle Based Sigmoid Formulation

The motivation behind the 3D formulation developed in this section was to better handle oblate bodies and deal with gnomonic projection, however let us assume momentarily that a spherical body is being imaged. This body already has an estimated location of its centroid in the field of view as well as an estimated angle of subtention. The subtended half-angle associated with the radius of this body is θ_r. Measuring from the apparent center of this body, any angle smaller than θ_r will be inside the visible body while any angle greater would be outside the body. Suppose there some pixels near the limb of the observed body, each subtending some angle θ_i with the center of the body. We can formulate the sigmoid function

$$S(\theta_i) = g_b + \frac{g_s - g_b}{1 + e^{k(\theta_i - \theta_r)}} \tag{3.12}$$

For pixels lying exactly on the limb, θ_i and θ_r coincide, and the sigmoid is at its midpoint, halfway through the graytone transition. Referencing Equation 3.12 it is clear that a graytone transition across the limb of an observed body can be described as a function of angular deviation away from the limb. However, this does not quite align with the mathematical framework which has been established as will be discussed in the next section.

A Comparison of ξ vs θ

In our triaxial ellipsoid framework, deviation from the limb is measured by parameter ξ, not by some angular deviation θ. The exact analytic relationship between ξ and θ has not been investigated, however numerical analysis has been performed.

If a relationship between the two values could be found then Equation 3.12 can be rewritten as Equation 3.13

$$S(\xi_i) = g_b + \frac{g_s - g_b}{1 + e^{kf(\xi_i)}} \tag{3.13}$$

where $f(\xi_i)$ maps ξ_i into an angular deviation. If such a formulation could be written, then the ξ values associated with every limb pixel from the selection band could easily be translated into an associated expected graytone value.

Before any investigation is done into the relationship between ξ and θ, it is already known that when the angular deviation of a vector from a limb is zero, i.e. when $\theta = 0$, ξ is also zero. This is the condition where Equation 3.10 is satisfied. This holds true for an observer at any position and an ellipsoidal body with any degree of oblateness.

One of the first investigations comparing ξ and θ involved placing an observer relatively close to a spherical body and defining a surface tangent vector which, from the observer's point of view' points towards what would be the observed limb. This vector is pitched through a range of angles. The angular range is set to go inward towards the center of the body by one quarter of the observed body radius and outward by the same amount. The width of this sweep corresponds roughly to the angle that would be subtended by the band of selected pixels which was prepared before nonlinear least squares. This scenario is shown in Figure III.15

At each step ξ is computed and the relationship between ξ and θ can be seen in Figure III.16. Though not mandatory, the angular range is intentionally defined so that a negative rotation points the vector away from the body, so that θ and ξ share the same sign. Note that while the sign of θ can be arbitrarily chosen — an

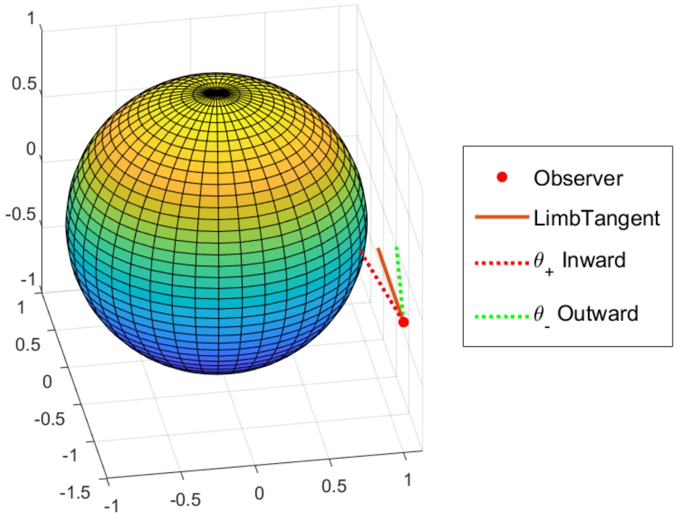


Figure III.15: An observer looking towards the limb of a spherical planetary body or moon. That observation is rotated through a range of angles towards and away from the body.

inward rotation towards the body can be defined as positive or negative and the same hold true for an outward rotation from the limb — this is not true for ξ. From numerical tests such as the one shown in III.16, it is evident that ξ is positive when angular deviation is inward towards the body, and negative when angular deviation is outward away from the body.

The somewhat uneventful results of Figure III.16 are encouraging, since they demonstrate that for the angular range under consideration, ξ is nearly linear with respect to θ. This is important, since the formulations for graytone transition across an edge which we have is based on angular measure. If the relationship between ξ and θ is linear, then only a simple scaling factor needs to be introduced when adapting a formula for describing the graytone transition to work in the ξ domain. These results

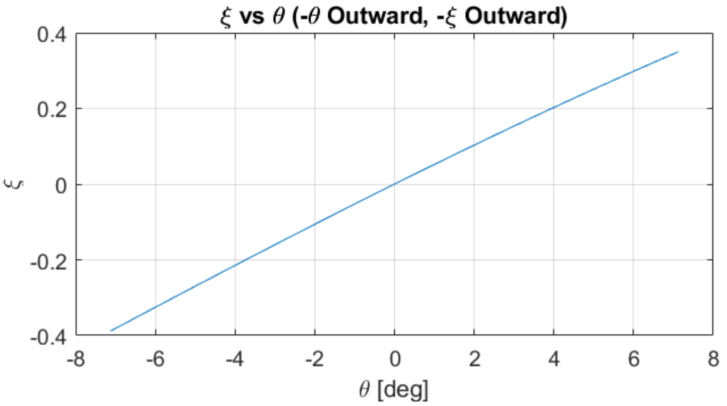


Figure III.16: Relationship between ξ and θ for an angular sweep across the limb of about 16 degrees. Note the relationship is nearly linear.

encouraged attempts to use a linear approximation to describe the relationship of ξ with respect to θ. Using a 3 point central finite difference method, the scaling factor between θ an ξ was determined to be about 101, when θ is described in radians. The approximation was tested at varying observer-to-body distances while bounding the range of θ to be one quarter of the half-angle subtended by the observed body. The results of one of these tests can be seen in Figure III.17.

At closest approach, the observer was placed to be 1.5 body radii away from the surface tangent point, roughly 1.8 body radii away from the center of the target body. At the farthest distances, the observer was 200 body radii away from the target body, so that the target body subtended only about 0.29 degrees. This corresponds to what the Moon would look like if the distance between the Earth and Moon were roughly doubled, since currently the Moon is about 100 Moon radii away from Earth. It was noted that, for close distances of a few body radii, the linear approximation was fairly good and slowly became worse as the observer moved away from the body.

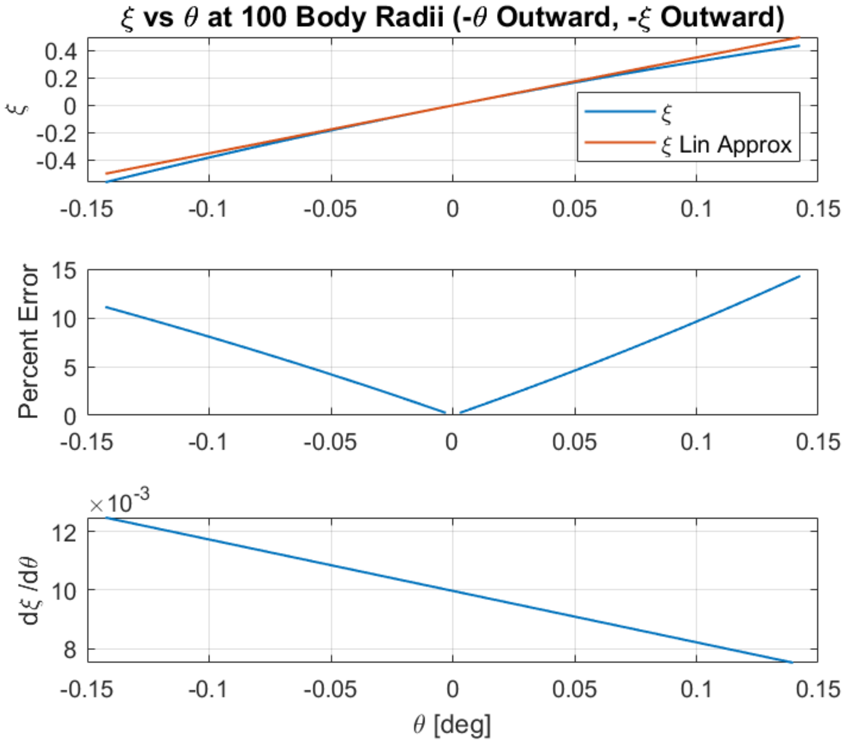


Figure III.17: ξ vs θ for an observer 100 body radii away from target body. **Top:** Linear approximation of ξ vs θ. **Middle:** Percent error of linear approximation. Discontinuity present at middle when both ξ and θ are zero. **Bottom:** Numerical derivative $d\xi/d\theta$

After about 4 to 6 body radii however the quality of the linear approximation stops changing. Whether an observer is 6 body radii or 100 body radii away from a target body the relationship between θ and ξ seems to remain fixed. Figure III.17 was generated at 100 body radii distance, however if that distance were changed, as long as it is greater than 6 body radii, the only thing that would change about the plots, is the angular range of θ. From the figure, it can be seen that the percent error of the linear approximation goes above 10% near the extrema of the angular range of

θ which indicates that a using ξ with just a constant scale factor may be insufficient for modeling the graytone transition across the limb. From the figure it is also clear that the derivative $d\xi/d\theta$ varies linearly over the angular range, indicating that a quadratic model would likely describe the relationship between ξ and θ well.

There was an additional concern that perhaps the simple rotation which pitches the observation vector towards or away from the center would not span all potential observation scenarios. To address this concern, a second orthogoanl rotation was added. This however, would be no different than choosing a different limb point, directly under the new tip of the observation vector, and simply pitching away from the center until the new tip location is reached. The same effect can be reached by also pitching the observation vector away from a target body and then simply rotating that body. What this thinking leads us to conclude is that the pattern seen in Figure III.17 should hold true for any point on the surface and that ξ should decrease uniformly in every direction as an observation vector is pointed away from the limb. This is supported by Figure III.18.

The final investigation into the relationship between ξ and θ looks into the effects of the oblateness of the observed body. Multiple tests were performed with bodies with varying oblateness, varying the semi-axes of $\mathbf{J}$ according to the sampling $a, b, c \sim U(0.5, 2)$. No variation in the relationship between ξ and θ is discernible, especially once the observer is more than a few body radii away from the target body. Figure III.19 shows the results from three different runs with bodies of different oblateness with the observer placed at 50 body radii away from the target.

There is no apparent variation in the error of the linear approximation of ξ

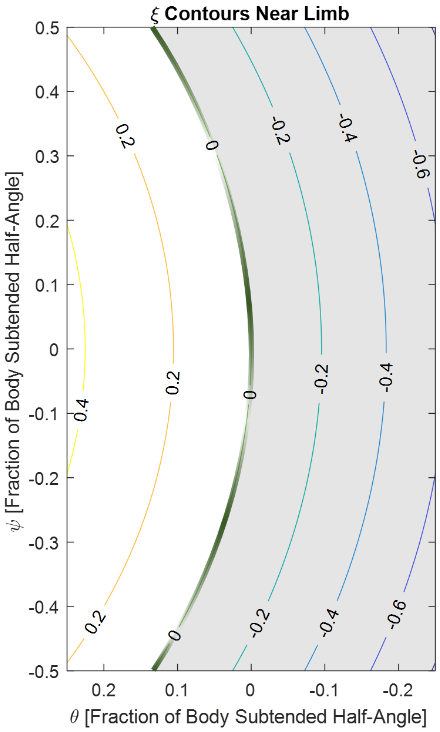


Figure III.18: Contours of ξ near the limb of an observed body. Angles are measured as fraction of the angle subtended by the body radius. ξ values on the outside of the limb are negative and all values inside are positive.

as a function of θ across the three different cases. While a full Monte Carlo test may be warranted, it is likely that a single mapping from ξ to θ would be sufficient for all observation cases. The remaining development of the nonlinear least squares formulation assumes this single mapping for all cases.

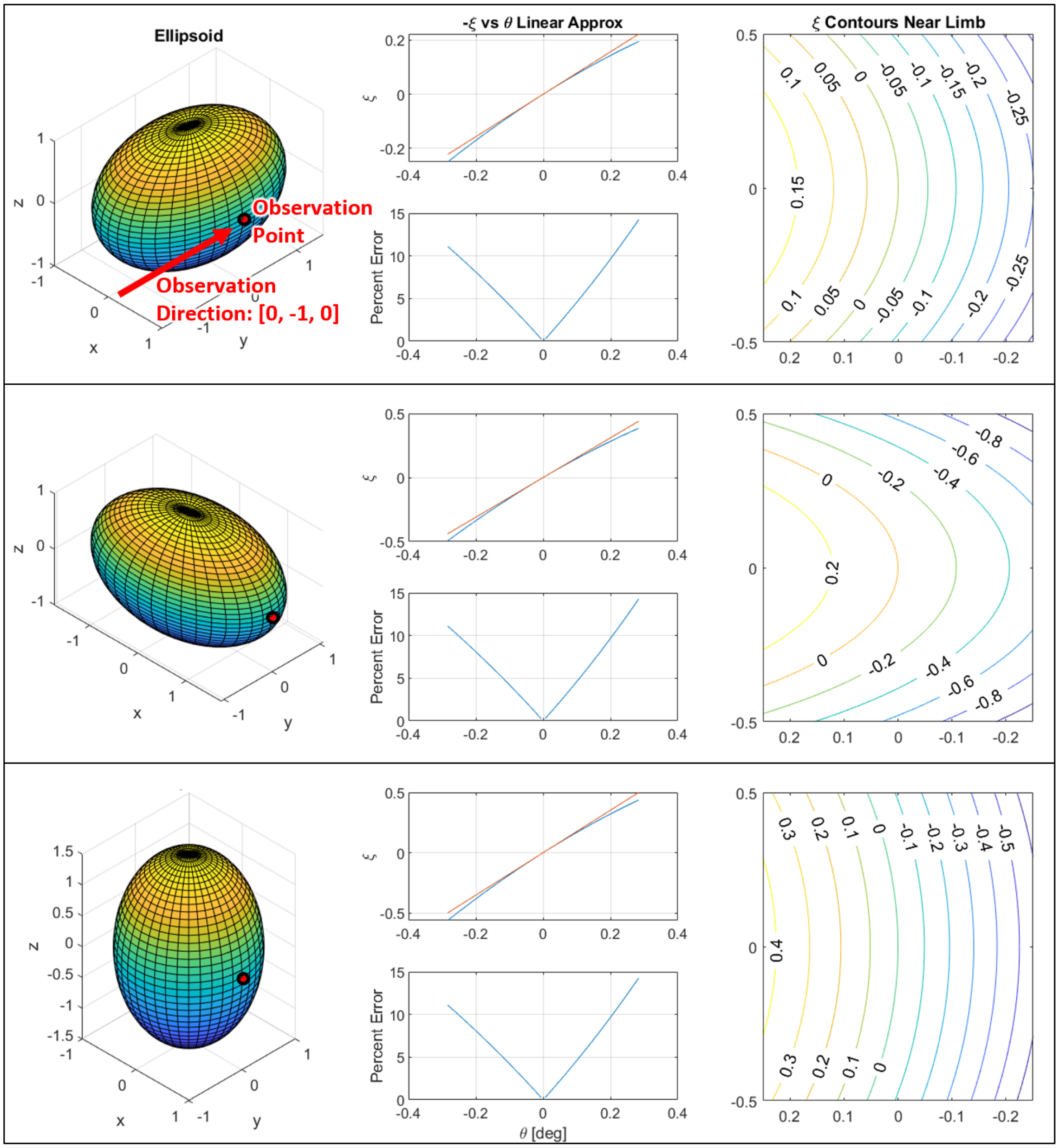


Figure III.19: Results from three ξ vs θ relationship tests with three ellipsoids of varying oblateness. All are imaged from the same direction from an observer located 50 body radii away from the target. Not the invariance of error for the linear approximation of ξ for the three different cases. Some axis labels have been omitted for clarity.

Final Conversion from ξ to θ, Cost Function Construction, and Iterative Update

Following the results of the analysis of the relationship between ξ and θ from the previous section, we adopt the following linear approximation of θ as a function of ξ

$$\theta(\xi) \text{ rad} = a \text{ rad } \times \xi \approx 101\xi \tag{3.14}$$

where a is a scalar constant determined to be about $a = 101$ using a 3 point finite central differencing technique. Combining Equation 3.14 and Equation 3.13 yields

$$S(\xi_i) = g_b + \frac{g_s - g_b}{1 + e^{ka\xi_i}} = g_b + \frac{g_s - g_b}{1 + e^{\kappa\xi_i}} \tag{3.15}$$

where the a can either be expressed as 101 or incorporated into κ. The associated cost function is

$$L_i = S_i - g_i = \left(g_b + \frac{g_s - g_b}{1 + e^{\kappa\xi_i}}\right) - g_i \tag{3.16}$$

Just like in the two dimensional case, g_b, g_s, and κ are relevant parameters which need to be estimated, however instead of a centroid and radius estimate there is instead $\xi_i = \xi_i(x, y, z)$ which has the ellipsoid body geometry $\mathbf{J}$ and the observer position $[x, y, z]$ built into it. The list of final parameters which must be estimated is $\mathbf{u} = [g_b, g_s, \kappa, x, y, z]$. Each of these parameters must be given an initial guess value

- g_{b0}: This represents the graytone of the surrounding background behind the body. It can be initialized as the average graytone of the pixels on the outer edge of the selection band.

- g_{s0}: This represents the graytone of the surface of the observed body. It can be initialized as the average graytone of all the pixels on the inner edge of the selection band.

- κ_0: For the two dimensional case, κ_0 was set to 1, here since the a factor is built into it, κ_0 can be set to 101.

- x_0, y_0, and z_0: The individual component of the initial position estimate given in the principal target body reference frame. From initial processing, a position estimate with respect to the observed body is provided in the camera reference frame. This position estimate is then expressed in an inertial frame using the known attitude of the observing spacecraft camera. This must now be expressed in the reference frame aligned with the principal axes of the observed target body. Knowledge of the orientation of this frame requires having ephemeris and a trusted clock available. With all of these elements combined, the initial position estimate is:

  $$[x_0, y_0, z_0]^{\mathrm{T}} = \mathbf{p}_{0,B} = C_{BI} C_{IC} \mathbf{p}_{0,C}$$

  where $\mathbf{p}_{0,C}$ is the initial position estimate with respect to the target body defined in the camera frame, C_{IC} is the camera-to-inertial direction cosine matrix (DCM) which rotates the camera frame into the inertial frame, and C_{BI} is the inertial-to-body DCM.

As before, we define

$$L_i = L_i(\mathbf{u}) \quad \text{with} \quad \mathbf{u} = [g_b, g_s, \kappa, x, y, z]$$

Linearizing L_i with respect to parameters $\mathbf{u}$ produces:

$$\begin{aligned} L_{i,k+1} =& L_{i,k} + \frac{\partial L_i}{\partial g_b}\Delta g_{b,k+1} + \frac{\partial L_i}{\partial g_s}\Delta g_{s,k+1} + \frac{\partial L_i}{\partial \kappa}\Delta \kappa_{k+1} \\ &+ \frac{\partial L_i}{\partial x}\Delta x_{k+1} + \frac{\partial L_i}{\partial y}\Delta y_{k+1} + \frac{\partial L_i}{\partial z}\Delta z_{k+1} \end{aligned}$$

The partials for this equation with respect to x, y, z are listed in Appendix A. The linearized equation which must be solved in order to calculate update $\mathbf{\Delta u}$ is thus

$$\mathbf{0} = \mathbf{L}_k + \mathcal{J}_k \mathbf{\Delta u}_{k+1} \tag{3.17}$$

Where $\mathcal{J}$ is the Jacobian:

$$\mathcal{J} = \begin{bmatrix} \vdots \\ \frac{\partial L_i}{\partial \mathbf{u}} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \\ \frac{\partial L_i}{\partial g_b} & \frac{\partial L_i}{\partial g_s} & \frac{\partial L_i}{\partial \kappa} & \frac{\partial L_i}{\partial x} & \frac{\partial L_i}{\partial y} & \frac{\partial L_i}{\partial z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

and the least squares update equation is:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{\Delta u}_{k+1} \quad \text{where} \quad \mathbf{\Delta u}_{k+1} = -\left(\mathcal{J}_k^{\mathrm{T}} \mathcal{J}_k\right)^{-1} \mathcal{J}_k^{\mathrm{T}}\, \mathbf{L}_k \tag{3.18}$$

The nonlinear least squares iteration is repeated until the stopping criteria

$$||\mathbf{\Delta u}_{k+1}||_2 < \varepsilon$$

is satisfied, or a maximum number of iterations is reached. The value of ε must be empirically determined depending on user needs.

To get the partials needed for populating the Jacobian $\mathcal{J}$ there are several intermediate partials which are needed, namely $\frac{\partial L}{\partial \xi}$, $\frac{\partial \xi}{\partial x}$, $\frac{\partial \xi}{\partial y}$, and $\frac{\partial \xi}{\partial z}$. Recall that $\xi = \mathbf{v}^{\mathrm{T}}\mathbf{M}\mathbf{v}$ and that $\mathbf{M} = \mathbf{J}\mathbf{p}\mathbf{p}^{\mathrm{T}}\mathbf{J} - (\mathbf{p}^{\mathrm{T}}\mathbf{J}\mathbf{p} - 1)\mathbf{J}$, so these partials of ξ in turn require the matrix partials $\frac{\partial \mathbf{M}}{\partial x}$, $\frac{\partial \mathbf{M}}{\partial y}$, $\frac{\partial \mathbf{M}}{\partial z}$. All of these partial derivatives can be found in the Appendix A, starting with Equation A.2, defined for generic ellipsoids, as well as special cases of ellipsoids with symmetries.

Position Estimate

One of the benefits to using this 3D approach over the 2D approach is that a position estimate in Cartesian coordinates is directly produced by the nonlinear least squares. If it is sufficient to define the position of the observer in the principal axis frame of the observed body, the values of *x,y,z* produced by the nonlinear least squares can serve as a final position estimate. If it is necessary to define it in an inertial or other frame, then the result can be expressed in whatever frame is necessary assuming the ephemeris for the body is provided. If the position of the observer is needed with respect to some other reference, then it is only necessary to know the position of the observed ellipsoid with respect to that reference, and the position estimate can be translated.

Method Limitations

During one of the tag up meetings with the Orion team at JSC a plot of the least squares residuals associated with the sigmoid fit of each Moon limb pixel was presented and It was noted that the residuals seemed to exhibit multi-modal behavior. There were at least two distinct groupings visible in the magnitude of residuals. This hurt the accuracy of the radius and centroid estimate so it was worth investigating what could be causing this bi-modal behavior.

After a brief discussion it was hypothesized that variations in lunar terrain were creating the observed behavior. The transition across the limb of the Moon into space is one of light to dark and the exact nature of that transition changes depending on how bright the lunar side is. The Moon is covered in highlands as well as seas, or *maria* in Latin, which are large basaltic plains that appear significantly darker. Referencing back to Figure III.8 one notices that the *maria* come up close against the lunar limb and are prominent when viewing the lunar nearside from Earth. They give the Moon it's distinct features which we recognize as the "Man in the Moon". It was believed that the lower graytone values associated with the *maria* led to a shift in the location for the sigmoid fits near them as shown in Figure III.20, leading to different beliefs about the location of the limb.

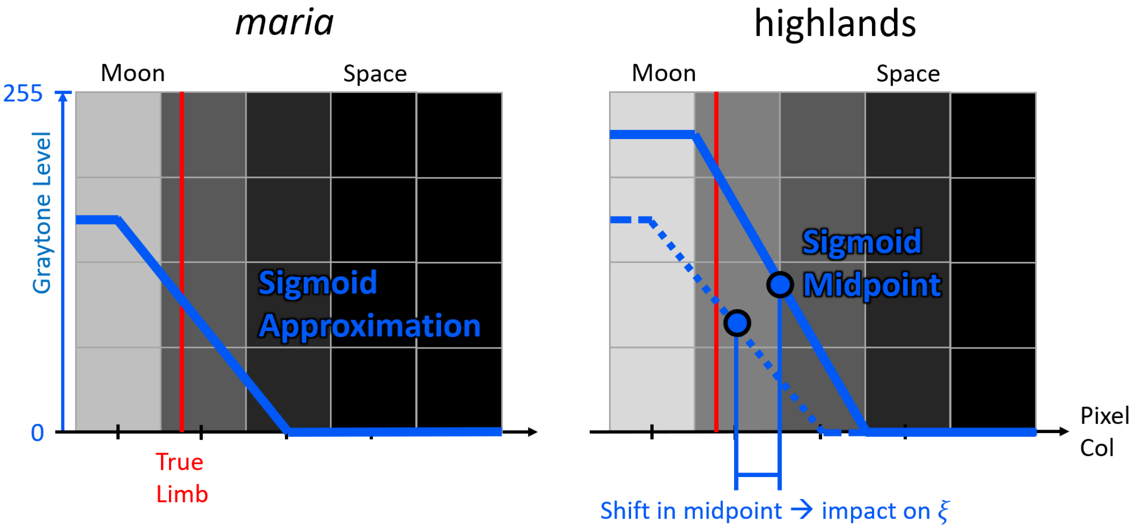


Figure III.20: Difference in apparent location of limb due to variation in brightness of surface of observed object.

The fact that a brighter object can have edges which appear to expand outward into the image scene shows that a sigmoid description of limb transition is not exact. In theory, a brighter object would simply have a larger sigmoid level step and a steeper transition, without shifting the location of the transition midpoint. However, in practice with digital optical systems, this is not the case. Brighter objects tend to have edges which glow, or expand out, into the scene. This is a complex phenomenon related to not only how light behaves in an optical system, but also to discretization of graytone values when mapped onto a digital imager. The effect is particularly noticeable for images which contains high contrast scenes where pixels become saturated. Once a body appears saturated in an image, the pixels associated with that body cannot become any brighter, so if the exposure is increased, the object will start appearing bigger, an effect known as “bleeding”. The appearance of the lunar highlands producing a slightly larger looking Moon is a less extreme expression of the same effect.

It turned out that variations in lunar terrain where creating different distance estimates. The *maria* were indeed shifting the estimated location of the limb. This problem could potentially be addressed if there was a way to “flatten” the surface of the Moon so that variations in topography and lighting do not have an impact on the image processing algorithms. The approach to resolving this problem is presented in the next section.

3D Gaussian Nonlinear Least Squares for Position Estimation

To resolve the issue presented in the previous section regarding the variations in lunar surface brightness leading to inconsistent distance estimates it was proposed to perform the limb fitting using a Gaussian fit on the image gradient, as opposed to a sigmoid fit on the original image. This approach worked well and was adopted as the final form for the nonlinear least squares position estimation algorithm. It has been described by Borissov [40], where the three dimensional Gaussian approach for limb fitting is presented, however is missing a functional relationship between ξ and θ which was developed in the previous section and used here.

Working on the Image Gradient

The gradient of the Moon images was already calculated at an earlier stage when performing the high-contrast pixel selection during the initial limb fitting. Referencing Figure III.21 one can look closely at the gradient of the limb and see the transition across it can be modeled by a Gaussian curve.

One can also note than, even though there is some noise on the inside of the lunar body, over the graytone of the body is relatively uniform. The variations in brightness associated with the varying lunar terrain do not exhibit high contrast and thus appear only faintly in the image gradient. Thus, the surface of the Moon appears more uniform in the image Gradient than in the original image.

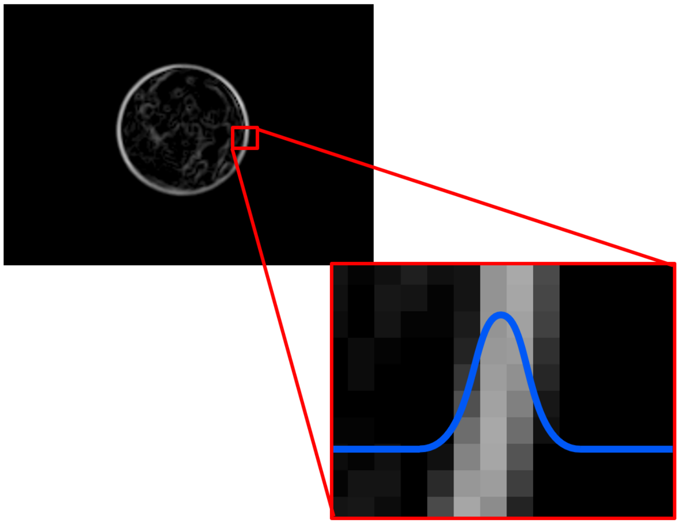


Figure III.21: Gaussian approximation of transition across gradient of limb.

Mathematical Formulation of Limb Fitting and Graytone Transition Across a Gradient 3D Scene

It stands to reason that since we are working with the image gradient, that our fitting function would be the derivative of a sigmoid, and indeed a Gaussian is the derivative of a sigmoid. When the gaussian function is used to model a normal probability distribution it appears as

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

In probability distributions, the peak hight of the Gaussian is normalized so that the area under the curve is 1. For the purposes of limb fitting, this constraint can be relaxed so that the peak hight of the Gaussian instead corresponds to the graytone of the brightest gradient limb pixels. Also, instead of defining the Gaussian as a function of distance from a mean, i.e. $x - \mu$, we can instead define it as a function of angular deviation $\theta_i - \theta_r$ or, like in the last section, as a function of ξ_i. Thus, the form of the Gaussian function which will be fitted to the limb gradient is captured in Equation 3.19.

$$g(\xi_i) = g'_{\max} e^{-\xi_i^2/\left(2\sigma^2\right)} \tag{3.19}$$

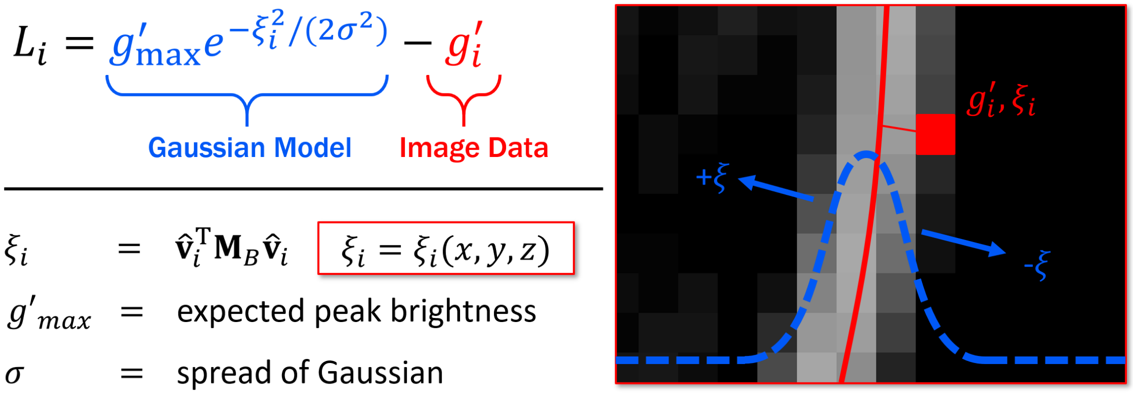
where $g'_{\max}$ is the peak graytone of the limb gradient and σ determines the spread of the Gaussian. If the linear approximation for $\xi(\theta)$ is used, then the scaling constant between θ and ξ can be embedded into σ.

Cost Function Minimization and Iterative Update

The cost function associated with the Gaussian fit is

$$L_i = g'_{\max} e^{-\xi_i^2/\left(2\sigma^2\right)} - g'_i$$

where g'_i is the graytone of the gradient pixels from the selection band as can be seen in Figure III.22.



$$L_i = \underbrace{g'_{\max} e^{-\xi_i^2/(2\sigma^2)}}_{\text{Gaussian Model}} - \underbrace{g'_i}_{\text{Image Data}}$$

$\xi_i \quad = \quad \hat{\mathbf{v}}_i^{\mathrm{T}} \mathbf{M}_B \hat{\mathbf{v}}_i \quad \boxed{\xi_i = \xi_i(x, y, z)}$

$g'_{max} \quad =$ expected peak brightness

$\sigma \quad =$ spread of Gaussian

Figure III.22: A summary of critical equations and concepts regarding Gaussian fitting on the limb gradient.

Looking to Equation III it is evident that L_i is a function of the unknown parameters $g'_{\max}, \sigma$ and ξ. As before ξ is a function of the position estimate x, y, z, so the final list of parameters which must be estimated through least squares is $\mathbf{u}$=$[g'_{\max}, \sigma, x, y, z]$. Initial values for $\mathbf{u}$ are:

- $g'_{\max,0}$: The limb of the Moon is a relatively high contrast edge, thus the gradient of edge pixels will be very bright. If the maximal possible graytone that an image pixel can take on is normalized to 1, then a good initial guess for $g'_{\max}$ is a value between 0.7 and 0.9. One can also collect the highest gradient pixels that were used during the initial limb fitting and use their average graytone value as an initial guess.

- σ_0: Taking a cue from statistical analysis, we can assume that the majority of the transition is described within the 3-σ bounds of the Gaussian. For many images, it takes about three pixels on either side of the limb for the gradient of the edge transition to fade to black. Thus, a reasonable initial guess for σ is one pixel. However, our cost function is not defined in the domain of pixels, but instead on the domain of ξ, thus pixel units must be converted to ξ. This conversion is done assuming pixels near the optical axis. Pixels far from the optical axis subtend a smaller angle than those near it, however for the purposes of getting an initial estimate for σ this is not relevant. A pixel near the optical axis subtends an angle $\theta = \arctan(d/f)$ where d is the size of the pixel and f is the focal length of the lens. For the nominal Aptina imager proposed for Orion, the pixel pitch is $d = 0.0022$mm and the nominal focal length is $f = 16$mm. Thus, $\theta = \arctan(0.0022/16) = 1.38e - 4$. Multiplying this by the scale factor 101 produces an estimate for σ scaled to the ξ domain: $\sigma_0 = 0.014$

- x_0, y_0, and z_0: The initial position estimate is the same as that for the sigmoid approach. It is reiterated here for ease of reference. The individual component of the initial position estimate given in the principal target body reference frame. From initial processing, a position estimate with respect to the observed body is provided in the camera reference frame. This position estimate is then expressed in an inertial frame using the known attitude of the observing spacecraft camera. This must now be expressed in the reference frame aligned with the principal axes of the observed target body. Knowledge of the orientation of this frame requires having ephemeris and a trusted clock available. With all of these elements combined, the initial position estimate is:

  $$[x_0, y_0, z_0]^{\mathrm{T}} = \mathbf{p}_{0,B} = C_{BI} C_{IC} \mathbf{p}_{0,C}$$

  where $\mathbf{p}_{0,c}$ is the initial position estimate with respect to the target body defined in the camera frame, C_{IC} is the camera-to-inertial direction cosine matrix (DCM) which rotates the camera frame into the inertial frame, and C_{BI} is the inertial-to-body DCM.

As with the sigmoid formulation, we linearize L_i with respect to the unknown parameters

$$\begin{aligned} L_{i,k+1} =& L_{i,k} + \frac{\partial L_i}{\partial g'_{\max}} \Delta g'_{\max,k+1} + \frac{\partial L_i}{\partial \sigma} \Delta \sigma_{k+1} \\ & + \frac{\partial L_i}{\partial x} \Delta x_{k+1} + \frac{\partial L_i}{\partial y} \Delta y_{k+1} + \frac{\partial L_i}{\partial z} \Delta z_{k+1} \end{aligned}$$

The partials for this equation with respect to x, y, z are listed in Appendix A. The linearized equation which must be solved in order to calculate update $\mathbf{\Delta u}$ is thus

$$\mathbf{0} = \mathbf{L}_k + \mathcal{J}_k \mathbf{\Delta u}_{k+1} \tag{3.20}$$

Where $\mathcal{J}$ is the Jacobian:

$$\mathcal{J} = \begin{bmatrix} \vdots \\ \frac{\partial L_i}{\partial \mathbf{u}} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial L_i}{\partial g'_{\max}} & \frac{\partial L_i}{\partial \sigma} & \frac{\partial L_i}{\partial x} & \frac{\partial L_i}{\partial y} & \frac{\partial L_i}{\partial z} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

and the update equation is:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{\Delta u}_{k+1} \quad \text{where} \quad \mathbf{\Delta u}_{k+1} = -\left(\mathcal{J}_k^{\mathrm{T}} \mathcal{J}_k\right)^{-1} \mathcal{J}_k^{\mathrm{T}} \mathbf{L}_k \tag{3.21}$$

The nonlinear least squares iteration is repeated until the stopping criteria

$$||\mathbf{\Delta u}_{k+1}||_2 < \varepsilon$$

is satisfied, or a maximum number of iterations is reached. The value of ε must be empirically determined depending on user needs.

As was the case for the sigmoid derivation, to get the partials needed for populating the Jacobian $\mathcal{J}$ there are several intermediate partials which are needed, namely $\frac{\partial L}{\partial \xi}$, $\frac{\partial \xi}{\partial x}$, $\frac{\partial \xi}{\partial y}$, and $\frac{\partial \xi}{\partial z}$. Recall that $\xi = \mathbf{v}^{\mathrm{T}}\mathbf{M}\mathbf{v}$ and that $\mathbf{M} = \mathbf{J}\mathbf{p}\mathbf{p}^{\mathrm{T}}\mathbf{J} - (\mathbf{p}^{\mathrm{T}}\mathbf{J}\mathbf{p} - 1)\mathbf{J}$, so these partials of ξ in turn require the matrix partials $\frac{\partial \mathbf{M}}{\partial x}$, $\frac{\partial \mathbf{M}}{\partial y}$, $\frac{\partial \mathbf{M}}{\partial z}$. All of these partial derivatives can be found in the Appendix A, starting with Equation A.2, defined for generic ellipsoids, as well as special cases of ellipsoids with symmetries.

Brief Overview of Application and Results

The Gaussian fitting algorithms were tested on multiple synthetic and real image sets of both the Earth and Moon. The results of the majority of that testing are covered in the next chapter along with analysis on the accuracy of the method. For the purposes of this section only a brief summary of some results is offered in order to paint a picture of how the algorithms actually works. Referencing Figure III.23 it can be seen how the quality of the fit changes after the nonlinear least squares iteration is complete.

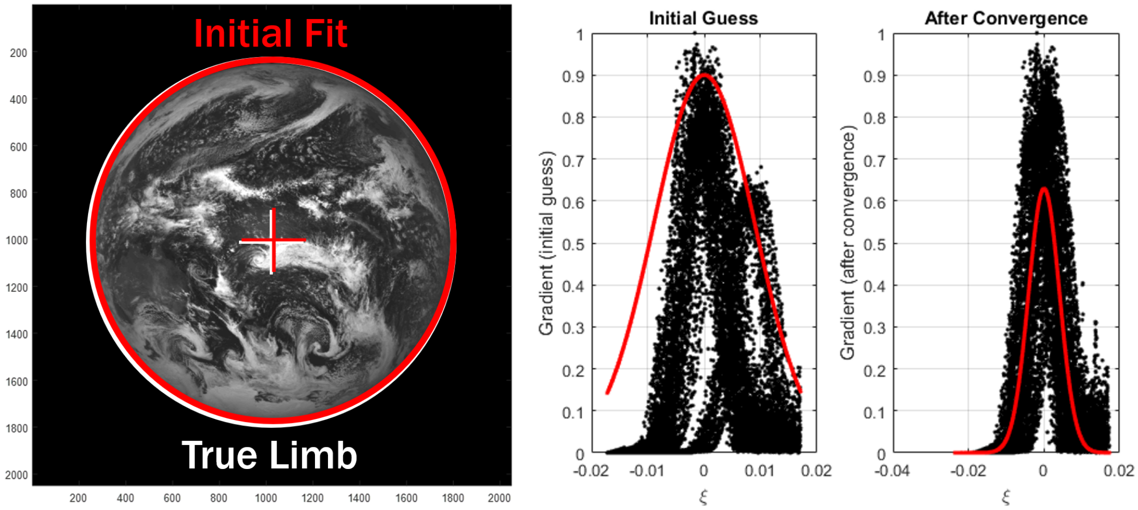


Figure III.23: Gaussian fitting applied on image of Earth taken from the DSCOVR Mission. Graytone values for all pixels selected for nonlinear least squares is shown as a function of ξ before and after nonlinear least squares. Note that even though the original image of the Earth is being shown, least square fitting was performed on the gradient.

Though the original image of the Earth is shown, nonlinear least squares was performed on the gradient. The bi-modal distribution in graytone that that can be seen in the initial guess is due to slight error in the initial centroid estimate for the Earth. Once this error was corrected through the nonlinear least squares, the majority of selected pixels align well with the Gaussian fit. It is not entirely clear why the band of pixels is as wide as it is. This could possibly be due to the high number of pixels which are being used. Since the original image was hi-resolution over a thousand pixels were selected for nonlinear least squares. Variations in atmospheric conditions which may also manifest itself as noise in the image which affects the graytone values of pixels near the limb. The initial guess of $\sigma_0 = 0.014$ wasn't too far off, as it appears the final converged value is somewhere close to 0.005 — about one third the starting value.

Method Limitations and Potential Future Work

A Gaussian fit using a 3D formulation for the nonlinear least squares offered improvements in performance and capabilities over the sigmoid formulation, however it does face some limitations, especially when applied to celestial objects with atmospheres. The atmosphere of planets or moons creates a softer limb edge than would be seen for terrestrial bodies. This "atmospheric glow" makes a planet or moon appear slightly bigger than it truly is which fools position estimation algorithms into thinking the observer is closer to the object than they really are. This effect was noticed when processing images of Earth. If the effect introduced a constant bias in distance estimation, then calibration could easily resolve the issue by estimating and subtracting out the bias. Unfortunately the bias introduced by this atmospheric glow is not constant. When one is close, this effect is more more pronounced and as an observer moves farther and farther away from an observed body with an atmosphere, the limb of the planet becomes more crisply defined and the atmosphere plays a smaller role. This variable bias introduces problems which require modeling a relationship between how the atmosphere appears as a function of observer to body distance.

Another potential limitation is the use of a linear approximation for $\theta(\xi)$. Switching to a more complex model, for example an inverse-quadratic model defining θ function of the square root of ξ would provide a better fit. However it is unclear if the added complexity of such a model is warranted. The error of the linear fit is less than five percent in the range $-0.05 < \xi < 0.05$ and looking at Figure III.23, they graytone values of the limb pixels drop down to near zero far before ξ reaches ± 0.05. It looks like most graytone values drop near zero at about $\xi = \pm 0.01$, where the linear approximation only incurs about one percent error. Thus, any potential gain in performance would likely be insignificant.

Under its current formulation, there is no measure of confidence or covariance which is associated with the position estimate provided by the nonlinear least squares. This makes it difficult to embed into a navigational filter as it is unclear how well the results are to be trusted with respect to other measurements and propagated position estimates. One potential source of uncertainty which affects the quality of the least squares fit and which can be characterized an modeled, is the consistency of the response of the pixel array. An imager would undergo a calibration process in which multiple images are captured of a fiducial marker under controlled but variable lighting conditions. With enough statistics gathered, the consistency with which each pixel responds could be modeled as a standard deviation in graytone as a function of the graytone. Thus, a confidence measure can be associated with every pixel for every graytone value. This uncertainty could then be propagated through the nonlinear least squares framework to give the uncertainty associated with the position estimate. Other potential sources of uncertainty which could be included could be inaccurate knowledge of the observed body geometry.

CHAPTER IV

OPTICAL NAVIGATION USING THE MOON AND EARTH - ALGORITHM TESTING, VERIFICATION, AND PERFORMANCE ASSESSMENT

One of the major challenges of developing the algorithms for Orion OpNav was figuring out a way of testing if the results were correct or not. Any developed algorithms would never be put on board a spacecraft, even just for testing purposes, without initial verification on the ground first. The difficult part about verifying position estimation algorithms for spacecraft navigation is the finding truth data. It is not sufficient to simply have pictures of the Moon taken from someone's backyard and feed them to the algorithm to see if it works. Such pictures were indeed used in the very nascent stages of algorithm development, however in order to verify that an OpNav position estimation algorithm works, it needs to be given pictures with location labels or tags. It's not enough to know where the picture was taken from with respect to some local reference, instead it must be possible to calculate the true position of the observer with respect to the Moon before image processing is done.

Finding such images was challenging, and so algorithm verification had to be broken down into phases. Method verification started with simple analytic tests of algorithm sensitivity, then moves on to testing with synthetic images, and eventually real images captured from space. In the first phase the goal was to demonstrate the sensitivity of core mathematical methods to error. The second phase tested the accuracy with which image processing algorithms could analyze images. The third phase tested algorithm susceptibility to noise and imperfections in the captured images.

Once the algorithms passed the series of tests they had to be-repackaged to run in core Flight Systems (cFS), a code architecture developed at Goddard Space Flight Center which simulates the flight systems of a spacecraft. The challenges of verifying that the developed methods work and preparing them for integration testing at Johnson Space Center are described in the following sections.

Initial Testing and Development in MATLAB

The first tests of the algorithms included sensitivity studies and Monte Carlo simulations using relatively simple synthetic images generated and analyzed in MATLAB. The results had to be compared against the accuracy requirements that had been provided by the Orion OpNav team.

Centroiding Accuracy Requirements

The Orion team at Johnson Space Center had one very specific requirement regarding the accuracy and precision of the image processing algorithms. It was stipulated that the standard deviation of the centroiding error had to be less than 0.1 Aptina Imager pixels. Centroid error was defined as the distance between the true and estimated centroid in the image plane. This means that if it were assumed that centroiding incurred no bias and error was distributed according to a nominal half normal distribution, after running centroiding tests on many images it would be required that roughly 60 percent of the error values are less than 0.1 Aptina pixels and that roughly 99 percent of error values are under 0.3 Aptina pixels.

This requirement stated in Aptina Pixels presented some unexpected challenges. While, from an operations perspective it might have made sense to define requirements in such a fashion it led to some difficulties later on when measuring algorithm performance. It is likely that at some point in the Orion team's design process it had been decided that a certain Aptina image sensor with a particular pixel size and particular lens stack with a particular focal length was selected for the project and this led to requirements being stipulated as such. This is however, conjecture and while it would have been preferable to have requirements provided in terms of angular pointing accuracy or position estimation accuracy, the Aptina Pixel requirement stayed and was something that had to be worked around. Unfortunately this meant that if a different sensor or different focal length was used in simulation or in testing the centroiding error results would have to be converted into Aptina pixels. Later version of the algorithms actually did not report estimated centroid and apparent Moon size, but directly provided a position estimate, which again had to be converted into Aptina pixels. However, the inherent shortcoming in defining requirements in terms of centroiding accuracy is that the same centroiding accuracy produces different position estimation accuracy at different distances from the Moon. This added another layer of difficulty when attempting to verify algorithm performance.

Sensitivity to Noise

One of the first set of tests that was run on the least squares centroiding algorithms was to analyze its sensitivity to shot and thermal noise. Random spikes corresponding to shot noise and dead pixels as well as random Gaussian noise was added to images of the Moon that had been captured from ground based telescopes. Sigmoid least squares fitting was then applied to these modified image sets as part of a Monte Carlo analysis to see whether the least squares fitting was robust to such noise and how much drift due to noise could be incurred performing the centroid estimate. Figure IV.1 shows the results of one Monte Carlo batch in which Gaussian noise with a standard deviation of 10 gray tone levels (on a scale of 0 to 255) was added to each image. The effect of this noise led to a centroiding error with a roughly Gaussian distribution with a 3-sigma of 0.1 pixels.

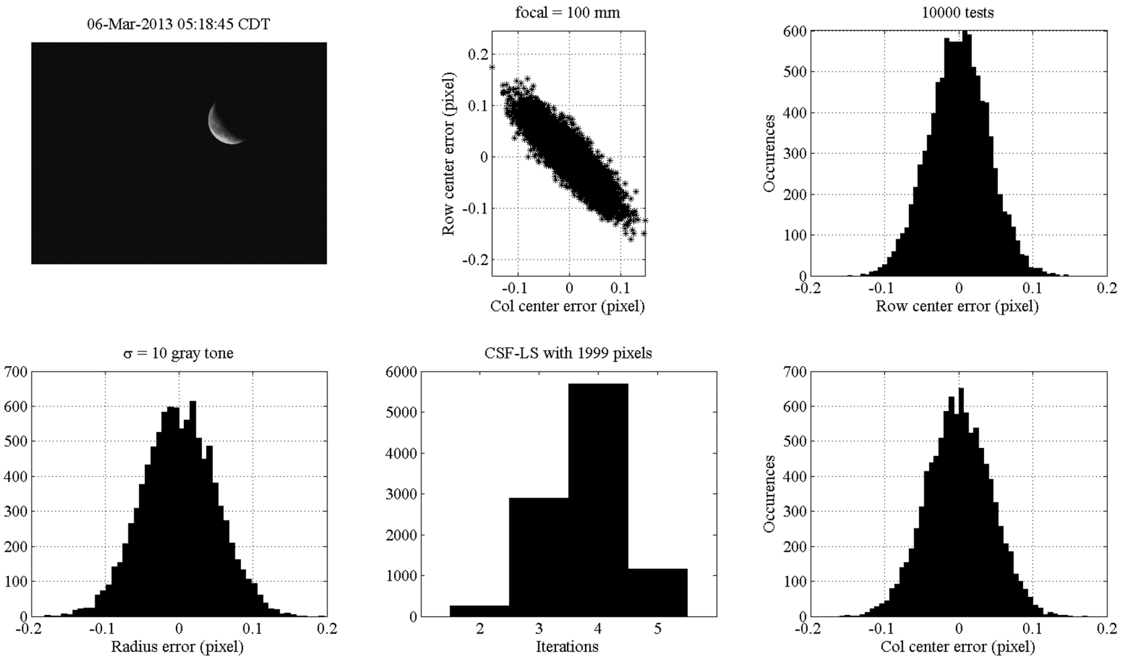


Figure IV.1: Results of Monte Carlo simulation for centroiding. Sensitivity of centroiding accuracy to image noise is shown.

In addition to ensuring that the algorithms were robust to noise, these tests showed that they work under drastically varied lighting conditions producing different Moon phases.

Sensitivity to Obliquity, Distance, and Focal Length

To expand the testing envelope of the algorithms a simulator was built which could simulate fully illuminated ellipsoids of varying obliquity, focus, focal length, and distance from the observer. First, it was necessary to get an idea of the relationship between focal length and apparent size of the Moon for a detector of fixed size and pixel pitch. This is captured in Figure IV.2.

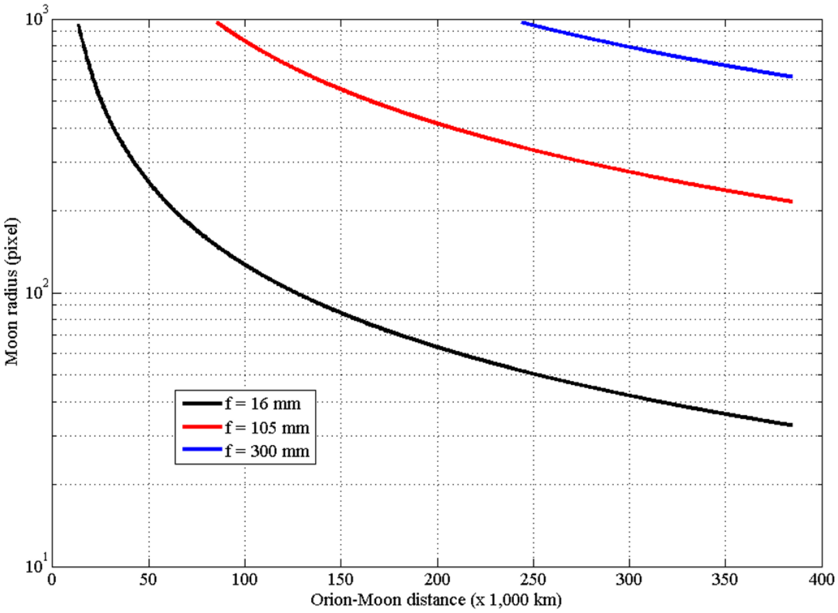


Figure IV.2: Apparent Moon size as a function of distance for three different focal lengths. A focal length of 16mm allows capture of the full Moon at distances ranging from 10,000 km to 400,000 km. 105 and 300 mm lenses cannot capture the full Moon at less than 90,000 and 240,000 km, respectively.

From this analysis it was evident what distance and focal length pairings could be used to create simulated Moon images. Due to gnomonic projection, the Moon (or any sphere) does not appear as a perfect circle when it is not centered in the camera field of view, but looks like an ellipse instead. This is especially pronounced for cameras with a wide field of view. Therefore, it made sense to be able to simulate oblate bodies. This also made the algorithm generalizable to any triaxial ellipsoid, such as the Earth. These bodies can appear as ellipses even when centered in the field of view. Finally, noise was added to stress the algorithms and add realism. Figure IV.3 shows a simulated Moon image with high distortion being passed to the sigmoid fitting function.

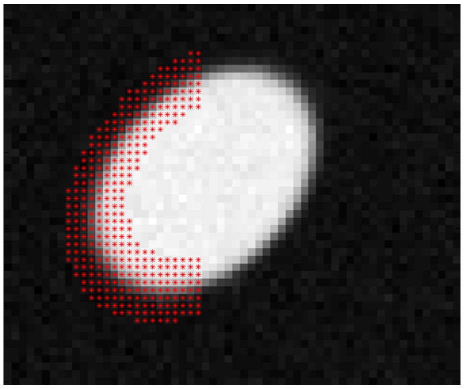


Figure IV.3: Example of Ellipse Sigmoid Fit pixel selection on simulated image with noise. Red marks indicate the pixels that will be used for least squares.

The goal here was, not necessarily to accurately characterize performance, but once more to prove feasibility and get an idea of the operational scope of the algorithms. Testing on these image sets indicated that algorithms could handle the Moon at varying distances and degrees of distortion.

Synthetic EDGE Images and Testing in the Electro-Optics Lab

Once basic feasibility was proven, the next phase of testing was dedicated to characterizing performance and understanding exactly what kind of precision and accuracy can be expected from the OpNav algorithms. The intuitive and obvious next step would be to go out on a clear night and capture some images of the Moon that can be used for testing. This approach however has several limitations. Since the OpNav algorithms would eventually be used on board Orion, or other spacecraft, in order to properly test the algorithms using realistic conditions, images taken from space where there is no atmospheric aberration and distortion are needed. This is especially important for understanding the stability of the algorithms when integrated representative hardware and used at relatively long distances to the Moon. One of the main issues, and one of the reasons why even minor atmospheric lensing or distortion can severely hamper position estimates, is the sensitivity of distance estimation to even the slightest change in apparent size of the Moon. This is not a significant problem when relatively close to a target object, such as the Moon, since changes in distance are paired with a fairly obvious change in apparent size. However, as shown in Figure IV.4, the apparent change in size which an observer notices when they move towards or away from an object, asymptotically approaches zero as their distance from the object increases. This means that even the slightest mis-perception of size can lead to enormous error in perceived distance when an observer is relatively far away from their target. Specifically, in regards to the appearance of the Moon when viewed from Earth, a change in apparent radius of a several arcseconds, a distortion which the atmosphere is more than capable of inducing, can lead to a change in estimated distance of a few thousand kilometers.

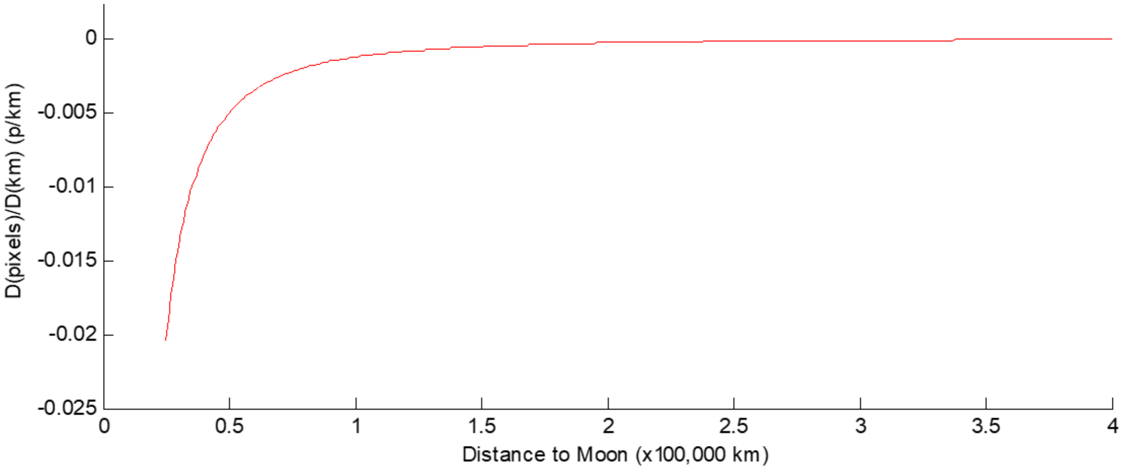


Figure IV.4: Sensitivity of apparent Moon radius as a function of distance to the Moon for a specific camera configuration. The change in radius is shown in units of pixel per kilometer.

This issues with sensitivity, and a desire for finding realistic images representative of the eventual operating environment, encouraged the use of space-based imagery of the Moon. However, even though it's possible to find images of the Moon captured by spacecraft above the Earth's orbit, these pictures by themselves are not enought to characterize the performance of the OpNav algorithms. To be able to verify that the algorithms can accurately estimate position when processing an image, it must be known exactly from where the picture was taken and at precisely what time, as celestial bodies move and thus a position fix without a defined epoch is not meaningful. Finding images which include that kind of meta-data is a difficult task, and while there are options available, such as images from the Lunar Reconnaissance

Orbiter (LRO) and the Lunar Crater Observation and Sensing Satellite (LCROSS), their availability is limited. Also, the available images might fail to stress the algorithms in ways which might potentially be encountered when used on mission, and thus it was decided that a simulation environment that generates realistic images of the Moon based on simulated spacecraft trajectories is needed.

The EDGE Rendering Environment

The chosen simulation tool which was used for rendering Moon images was NASA's Engineering DOUG Graphics for Exploration (EDGE) developed at Johnson Space Center. EDGE is a rendering environment capable of providing realistic textures, lighting, and model representation. Example rendering from EDGE can be seen in Figure IV.5. By itself, however it cannot simulate the underlying physics and dynamics which would be needed for a simulated cislunar trajectory. For that, NASA's Trick Simulation Environment is used to generate simulated spacecraft trajectories as well as tracking spacecraft position, camera attitude, and timing information.

With Orion's cis-lunar trajectory calculated and simulated in Trick by JSC engineers, EDGE image sequences could be generated which represent what Orion's camera's would see en route to the Moon coming from Earth. One of these preliminary image sequences is shown in Figure IV.6. Note that in this sequence there are images where the observer is too close to the Moon to be able to see the full lunar limb. In theory, once Orion were to get this close to the Moon a different form of navigation would be needed, most likely revolving around tracking of surface

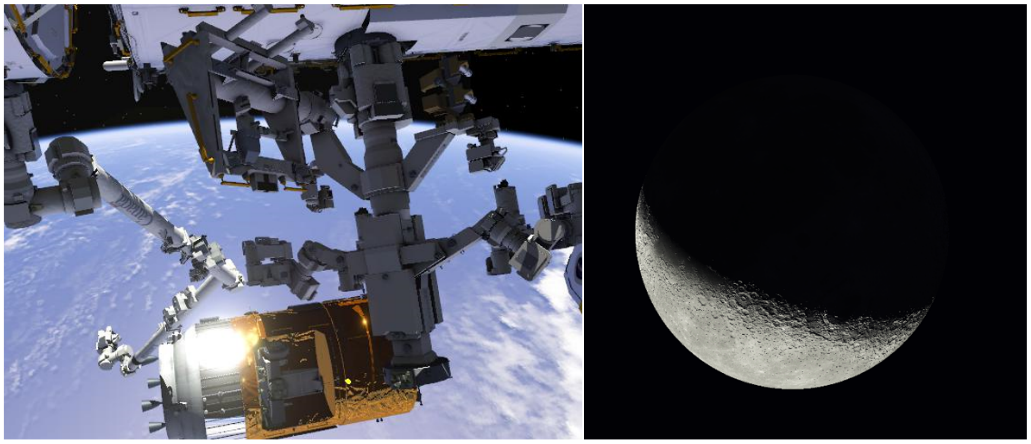


Figure IV.5: Renderings generated in EDGE.

features.

In order to expand the simulated operational envelope random, but known, pointing error was added to the camera's attitude on every frame so that the Moon doesn't always appear perfectly centered in the Field of View (FoV). Lighting conditions were also varied by randomly moving the Sun around so that different phases of the Moon were visible and at different lighting angles. The Sun's gravity was ignored in the simulations, so this has no effect on the trajectory. This variable lighting mimics the variability in appearance that one would see when travailing at different times of the lunar cycle and when the camera's roll angle is varied.

Lab Test Setup

Using EDGE by itself to generate images for testing, however was insufficient. While EDGE has some basic tools for defining different camera parameters, such as FoV or magnification, and includes some optical effects such as lens flares, it is not a camera simulator, so things like defocusing, distortion, thermal noise, and shot noise

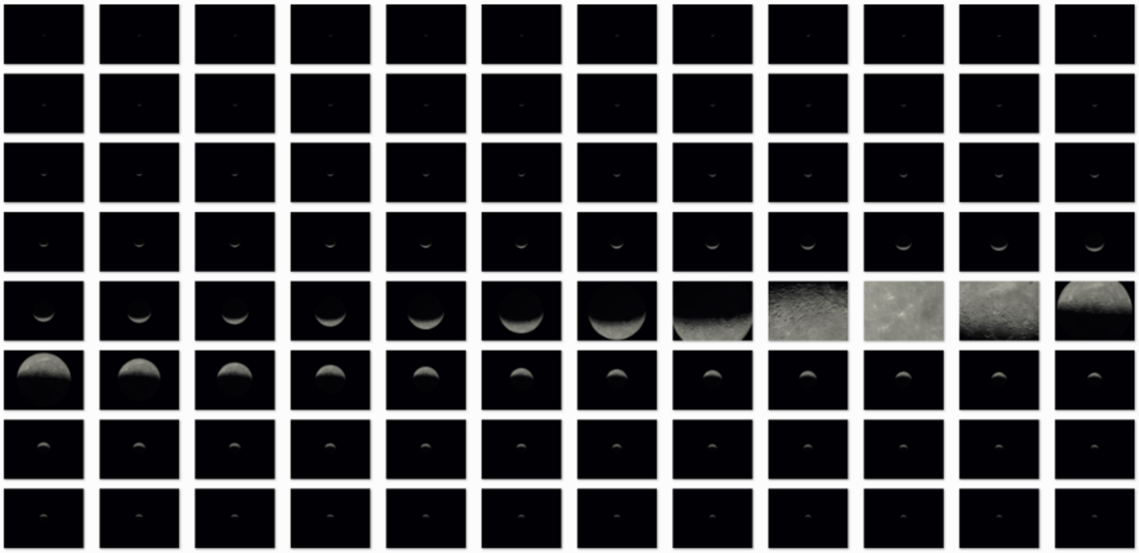


Figure IV.6: A sequence of images generated in EDGE showing the Moon from the point of view of Orion during lunar approach, orbit, and return to Earth.

are not simulated. Therefore, in order to increase the realism of the images, it was decided that a real camera would be used to capture images of the Moon rendered on a screen. The majority of algorithm development during the summer of 2014 was dedicated to setting up the experimental rig in the Electro-Optics Lab at JSC. Figure IV.7 shows a preliminary set up in the lab with a camera aimed at an EDGE rendering of the Moon being displayed on a 4K monitor.

The remainder of this section will focus on the EDGE rendering and preparation for algorithm testing. The initial portion of rendering was dedicated to setting up the correct EDGE environment and defining the cameras correctly. Once the EDGE environment was set up, and trajectory information was provided, the next task was to precisely align the camera with the screen. The simulated EDGE images were displayed as a slideshow controlled by a computer which was also in control of the

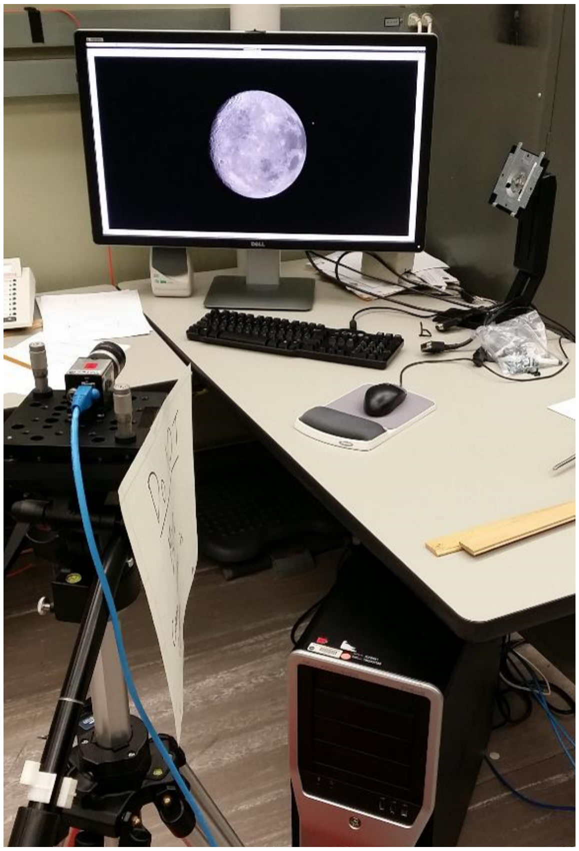


Figure IV.7: Preliminary camera calibration for capture of synthetic EDGE images in the Electro-Optics Lab at JSC. An EDGE rendering of the Moon is being displayed on a 4k display while a tripod mounted camera captures live footage.

camera, thus the image slideshow could be synchronized with the camera capture. Once all of these pieces were set in place, capture sequences such as the one shown in Figure IV.8 could finally be created.

These images captured in the lab would include some small amounts of optical distortion and noise effects. Thus, not only would they represent imaging conditions more realistically, they could also be used to test algorithm sensitivity to camera calibration. Optical system which must perform with a high level of precision must

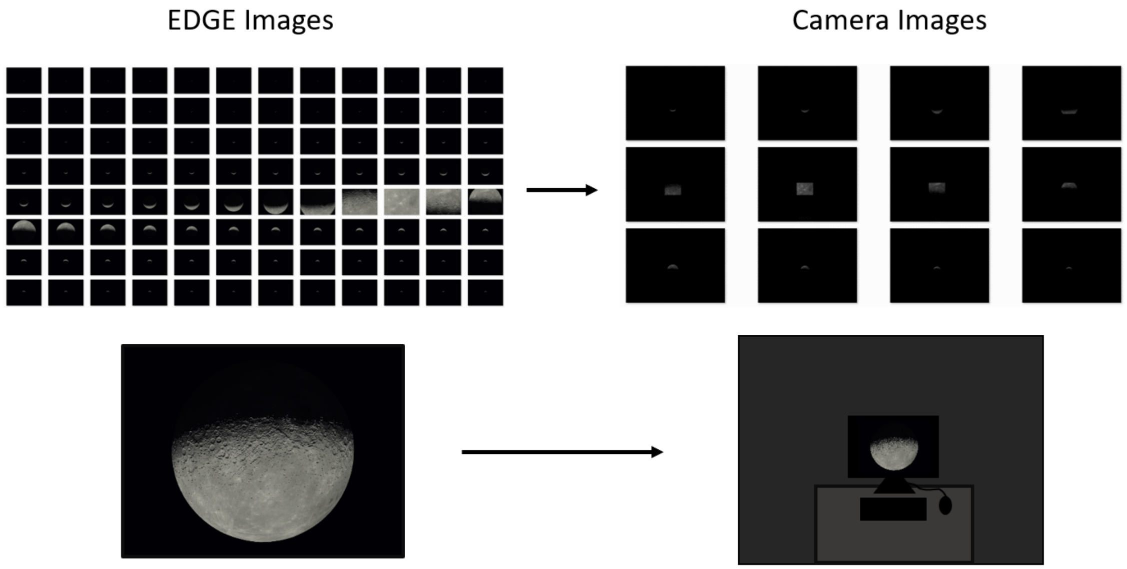


Figure IV.8: The translation of EDGE Moon images to Camera Moon Images.

be calibrated and this would apply to the camera system which eventually flied and operates on board Orion. Knowing how sensitive OpNav algorithms are to the quality of the calibration and characterization would inform future operators and engineers what procedures must be carried out on mission before the camera system can be used reliably as a backup navigation solution.

The final step necessary before processing was to crop the images, removing the black space surrounding the image shown on the monitor, and to calculate the effective FoV of the camera images. FoV or focal length parameters need to be provided as input to the OpNav algorithms in order to correctly process the images and provide a position estimate. The FoV associated with the lab images is neither the FoV of the camera, nor is it the FoV of the camera in EDGE used to render the virtual scenes. The effective FoV associated with the lab Moon images would be a function of how far away the screen is placed from the camera. Figure IV.9

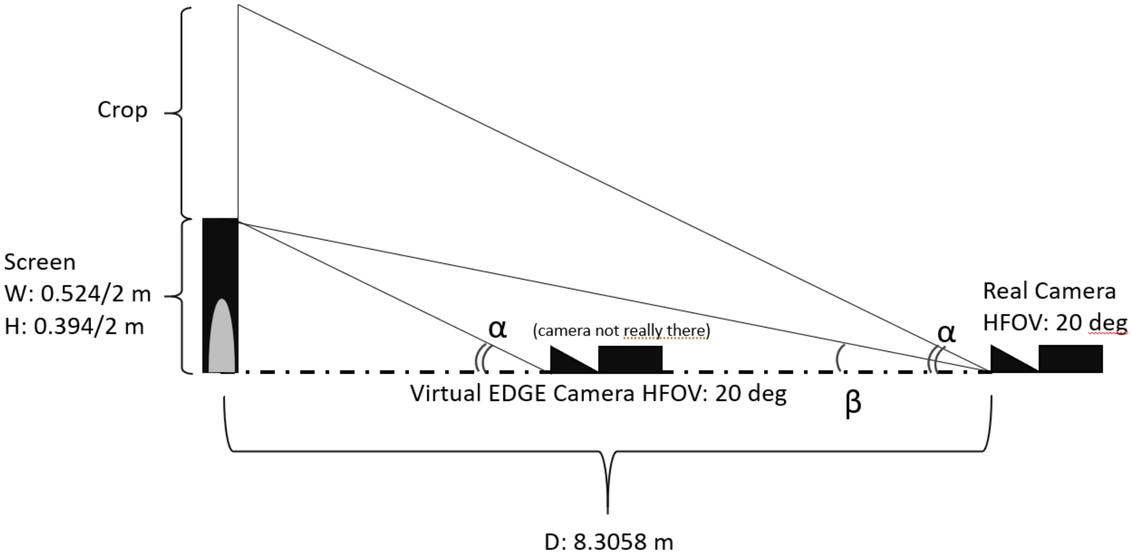


Figure IV.9: The effective FoV of the Moon camera images after cropping. Camera in lab and virtual EGDE camera both have a HFoV of 20 degrees.

shows what the effective half horizontal FoV (HFoV), β would look like for a specific lab configuration. With the lab configuration completed, all the necessary pieces for generating and analyzing synthetic lab images were in place for image capture and algorithm testing to commence. In addition to these lab images, other image sets were acquired from the ISS. The processing of these real images is covered in the next section.

Images Captured from the ISS and Overall Assessment of Performance

The image sequences produced by EDGE made it possible to roughly characterize the expected performance of the OpNav algorithms and perform sensitivity analysis. This analysis could be used to create a model that relates OpNav performance and accuracy to Moon distance. However, this testing could not provide an idea of what kind of accuracy can realistically be expected from OpNav when used on board a spacecraft with real imagery captured from space. To address this issue, the following summer, images of the Moon were captured by astronaut Terry Virts from on board the ISS and given to the Orion team for processing. With the image sets in hand, it was possible to estimate the precision and accuracy with which the algorithms could estimate the distance to the Moon. However, it should be noted that pointing accuracy or precision could not directly be estimated, since this would require precise knowledge of the attitude of the camera expressed in an inertial frame, and such information was not available.

Several image sets were captured from the various viewing ports available on the station. Figure IV.10 shows views of the Moon taken from the Cupola on board the ISS. Again, the goal was to present pictures of the Moon that were more representative of what might actually be observed. However, there were some additional difficulties which were not expected. Some of the viewing ports on the space station have protective screens on them which added slight distortion. Also, some of the image sets had adjustments made to the focus during the imaging session. While these effects are not necessarily representative of the eventual imaging camera, they still offered useful opportunities for stress testing the image processing algorithms.

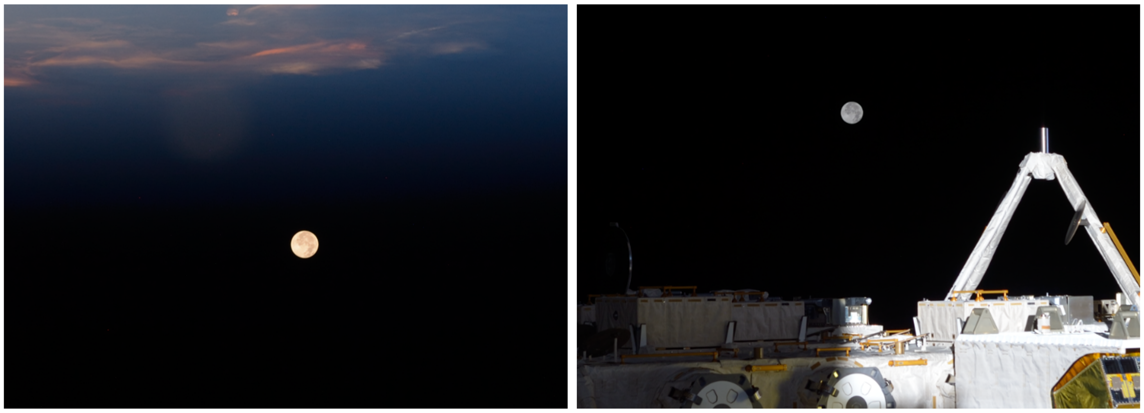


Figure IV.10: Two sample images showing the Moon viewed from the ISS. On the left the Moon has a reddish hue due to light passing through the Earth's atmosphere. On the right, structures from the ISS can be seen in the foreground. Photo Credit: Terry Virts, NASA

Another useful test was that of the algorithm's robustness to interference. While it is unlikely that an OpNav camera designed fro viewing planets would be mounted in such a way that parts of the Orion would be visible in the field of view, having portions of the ISS visible in the images allowed us to test how algorithms would fair in the case of such an eventuality. To make the algorithms work for such images, tweaks had to be made to limb pixel selection process so that edge pixels not belonging to the limb would be rejected.

For example, Figure IV.11 shows chromatic aberration as the slight difference in color on the left and right sides of the Moon. This could lead to a slight difference in brightness when converted to grey scale, leading to slightly differing estimates of distance. Other difficulties were introduced by the gamma correction which the Nikon camera which was used to take the photos automatically applies to images to suit the needs of human observers. However, such gamma corrections would not be applied to machine vision cameras such as the ones which would eventually be installed on Orion.

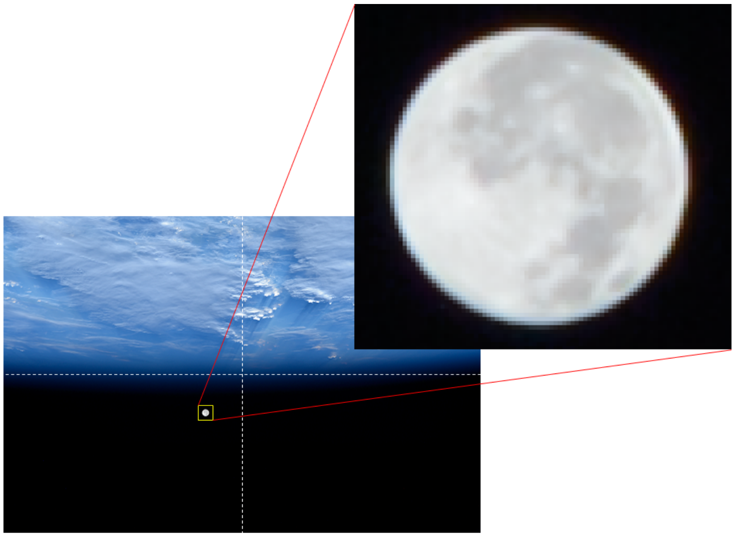


Figure IV.11: Chromatic aberration is visible on the limb of the Moon. The lower left side has a blue hue and the upper right side has a red tint. Photo Credit: Terry Virts, NASA

One of the benefits of having images with a known time stamp captured on board the ISS is that the precise location of the observer at the time of image capture is known. The ISS is tracked by ground stations and one can readily find a time series of its position in space by searching for the Two Line Elements (TLE) of “Zarya”, the Russian name of the ISS’s Functional Cargo Block, and the first module of the ISS to be launched. The time stamp of the images was also used to locate the position of the Moon using ephemeris from JPL Horizons online service. Thus, the true distance from the ISS to the Moon at the time of image capture could be calculated.

Six image sets, three pairs, were taken from the Cupola of the ISS and processed. The results of this processing have been previously presented by Borissov [41] and are reproduced here. Figure IV.12 shows how the estimated distance to the Moon compares to the true distance to the Moon in these six sets. The last pair of image sets show noticeably larger deviation from the trues distance compared to the first two sets and it is believed that this is due to a change in focus of the camera during image capture. This highlights the importance of having a well characterized and calibrated camera with fixed parameters which do not change if one desires to have the most accurate navigation solution possible.

It is clear that over the course of each imaging set the distance between the ISS and the Moon seems to decrease. This is due to the geometry and orientation of the Cupola. The Cupola on the ISS is meant to be an observation port for viewing Earth, and thus points nadir. It was chosen for imaging the Moon due to the better optical qualities of the Cupola windows compared to those of other ports, however due to its position on the ISS, the Moon could only be visible at Moon rise and Moon set, for a relatively short interval before it would pass out of view. Thus, all imaging sessions were done as the ISS was moving towards the Moon, right after Moon rise. This produces the noticeable decrease in distance over the course of the imaging window. Other image sets which were taken from other ports on the ISS do not exhibit this behavior but arc instead.

There is also a large change in distance between the three pairs of viewing sessions. This is due to the amount of time between each viewing pair, which was on the order of one month. Over the course of this time, the Moon's distance to the Earth changed by tens of thousands of kilometers. This distance can vary by over 50,000 km from perigee to apogee over the course of roughly two weeks, or one half a lunar orbital period. Within each pair of viewing sessions however, the two sessions were done one orbit apart, which corresponds to an interval of roughly 90 minutes. Over the course of these 90 minutes the distance from the ISS to the Moon at the start of the viewing time window does not significantly change.

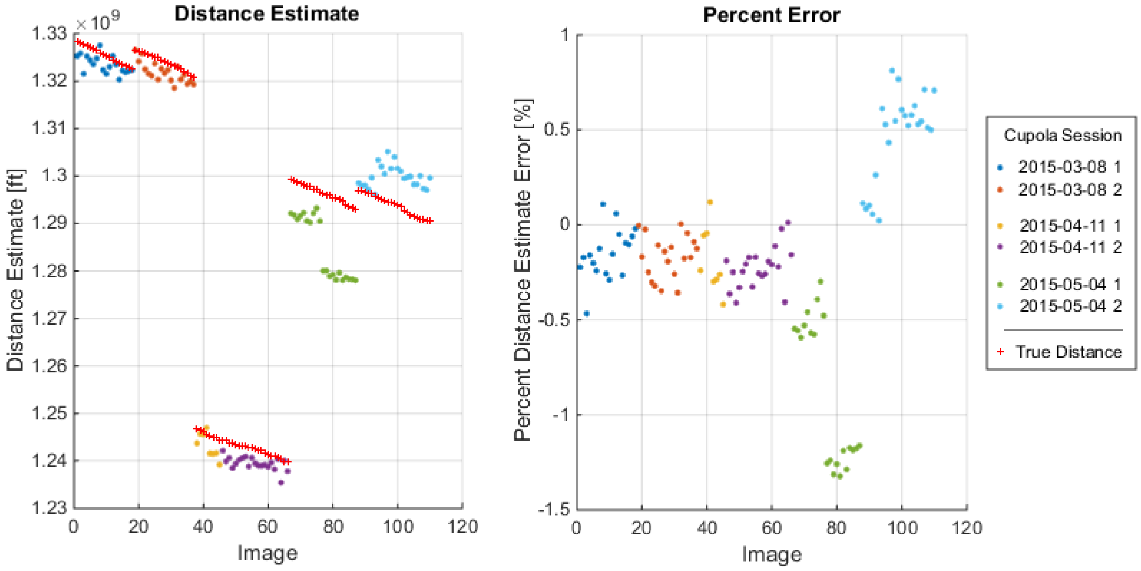


Figure IV.12: Initial Distance Estimation Error. Left plot shows estimated distance to the Moon for each image taken from the ISS and the right shows the corresponding percent error of that distance estimate.

Due to the inconsistencies present in the last two Cupola sets, they omitted when analyzing and characterizing the performance of the algorithms. Thus, the final evaluation of algorithm's performance on the ISS Moon images was based on the first four sets, which in total contained about 65 images.

On average, distance estimation error of the first 4 Cupola sets was an overshoot of about 717.0 km. Since any camera system which is deployed on mission would realistically go through a calibration procedure, it is reasonable to expect that a bias correction to the distance estimation would be applied, so that on average the error is zero. With this bias correction of 717.0 km subtracted, the RMS error, identical to standard deviation in the case of a zero mean, was 505.28km. This represents 0.128 percent of the total Earth to Moon distance. The results of this final analysis are shown in Figure IV.13. The two plots shown indicate that the error distribution is roughly Gaussian, and thus standard statistical analysis tools and jargon can be used to describe the error distribution.

As can be recalled from the beginning of the section regarding centroiding precision requirement, NASA had stipulated that algorithms must be able to perform centroiding to a precision of 1 sigma equaling 0.1 Aptina pixels. The ISS pictures however, were taken with a handheld Nikon camera with a different focal length and sensor size than the nominal Aptina imager. Also, the algorithms reported error in terms of distance to target object, not in terms of centroiding error on the sensor. So, to verify the requirement was met it was necessary to convert the distance error into a pointing error expressed first in degrees or radians, and then converted to Aptina pixels. Again, this was the only way to calculate centroiding or pointing error, since true camera attitude was not known.

To help understand how distance estimation error could be converted to pointing error, reference Figure IV.14 and Equations 4.1, 4.2, and 4.3.

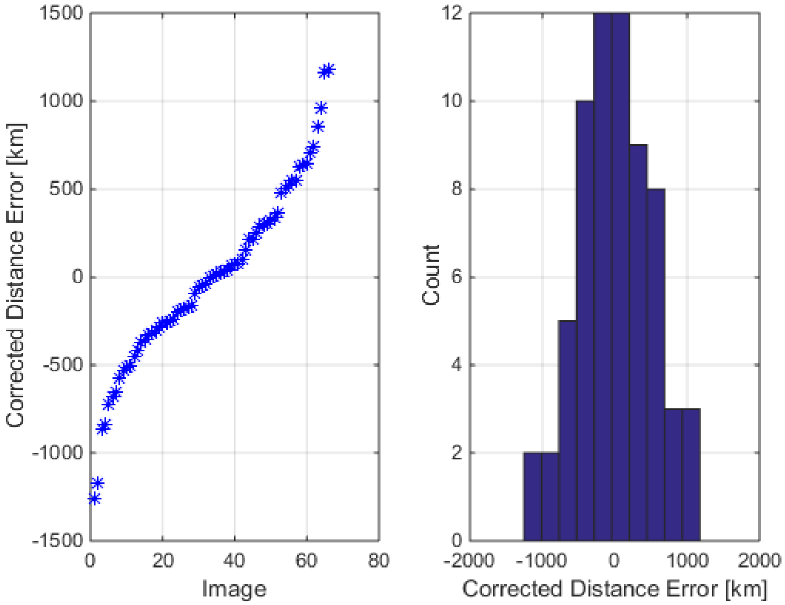


Figure IV.13: Bias Corrected Error. The left plot shows the bias corrected distance error sorted over the 4 image sets. The right shows a histogram of the boas corrected distance error

$$\frac{R}{D} = \sin\theta \qquad\qquad \frac{r}{f} = \tan\theta \tag{4.1}$$

Given the radius of the Moon R and the true distance between the observer and the Moon D, it is easy to calculate the expected subtended angle θ associated with the lunar radius using the first equation in 4.1. This subtended angle projects the lunar radius onto the image plane according to the second equation in 4.1, where f is the focal length. For the nominal Aptina imager, f is 16 mm. Note that θ is not dependant in any way on the camera system setup, but is purely a question of the geometry describing the observer and target. Thus if pointing error is known, it can be converted into pixel error for any camera configuration.

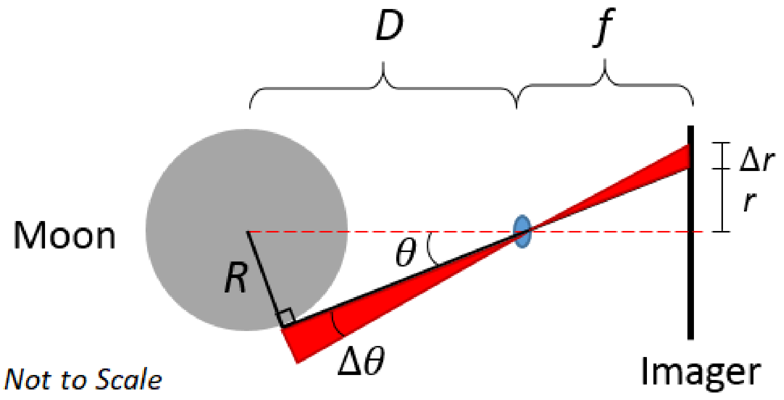


Figure IV.14: Pinhole camera model and the Moon. How angular deviation translates into linear deviation on the imager. Note that $D >> f$.

To convert error in distance estimate ΔD into pointing error $\Delta\theta$ Equation 4.2 is used.

$$\begin{aligned}\Delta\theta &= \sin^{-1}\left(\frac{R}{D+\Delta D}\right) - \sin^{-1}\left(\frac{R}{D}\right) \\ &= \sin^{-1}\left(\frac{R}{D+\Delta D}\right) - \theta\end{aligned} \tag{4.2}$$

This equation simply expresses the angular difference, $\Delta\theta$, between the estimated subtended lunar half-angle and the true subtended lunar half-angle, θ. Finally, Equation 4.3 is used order to convert the angular difference $\Delta\theta$ into a distance error on the imager plane. Δr will take on whatever units f is described in, which is nominally millimeters.

$$\begin{aligned}\Delta r &= f\tan(\theta+\Delta\theta) - f\tan(\theta) \\ &= f\tan(\theta+\Delta\theta) - r\end{aligned} \tag{4.3}$$

Potential Complications and Final Requirements Verification

The last step is to convert Δr from units of millimeters, to Aptina pixels. For the chosen sensor, the pixel pitch is 0.0022 mm/px. Dividing Δr by 0.0022 would give the desired effective pointing error described in Aptina pixels. There are however, a few potential complications that must be cleared up before verification of the algorithms can be complete.

The first complication is that both Equation 4.2 establish a dependancy between pointing error $\Delta\theta$ and distance to the Moon D. The distance from the ISS to the Moon is not constant and varies by roughly 6 percent between the imaging sessions. To ensure that this was accounted for, the pointing accuracy conversions were done using the ISS to Moon distance from the both the closest and furthest imaging distances. The effect on the final precision was a difference in precision on the order of 9 percent of the 0.1 pixel accuracy requirement, enough to be taken into consideration if algorithms were in danger of not meeting requirements. However, whether the closest of shortest distance is used in the calculations, the pointing error corresponding to a 505km distance estimation error is between 0.048 and 0.038 Aptina pixels. Both of these precision estimates are well within the required 0.1 Aptina pixel pointing precision requirement.

The other potential complication would be the sign of the Δ terms in Equations 4.1 and 4.2. Since sin and tan are nonlinear functions, a positive or negative deviation in input would produce different deviations in output. All possible permutations of the signs of the Δ terms were tested on no difference was noted in the final precision estimate when reporting th precision using 3 significant figures. Based on these findings, it can be concluded that the effect of the sign of the deviation on the final precision estimate is negligible when dealing with approximately Earth-Moon distances, and can thus be neglected.

To recapitulate, the 1-sigma pointing precision of the OpNav algorithms used on the ISS images was estimated to be about 0.04 Aptina pixels, more than double the required precision of 0.1 Aptina pixels. The requirement was verified and the results are actually better than they look! These results were obtained using a hand-held camera with variable focus taking pictures from behind the glass of the Cupola. Also, no camera calibration or characterization was done and the focal length of the camera was assumed to be the nominal 58 mm as specified by Nikon. Even better performance could be expected when using a dedicated machine vision camera with constant focus mounted in the vacuum of space that has gone through thorough calibration and characterization.

CHAPTER V

X-RAY PULSAR-BASED NAVIGATION - BACKGROUND, LITERATURE REVIEW, AND DEVELOPMENT FRAMEWORK

The study of pulsar based navigation has been growing in popularity as technological hurtles which stood in the way of it's implementation are being overcome. One of the largest technical innovation which has made the technology from theory to practice has been the development of X-ray antenna which are capable of collecting signals from X-ray pulsars. Unlike previous radio telescope based observation of pulsars, these X-ray sensors have a significantly smaller form factor which allows them to be placed on spacecraft. This innovation has led to a growing interest from both NASA through the work of Goddard's SEXTANT team [42] and China through their XPNAV-1 mission [43]. The promise of a universal navigation for spacecraft has piqued the interest of many research teams and stimulated research. in signal processing hardware and algorithms.

The Story of Algorithm Development

Investigations in X-Nav by our team began first with the question of how a star catalog would need to be updated if a spacecraft were to travel on an interstellar mission [44]. Parallax between stars would begin to be noticeable as a spacecraft leaves the solar system, and thus cataloged angles between stars would need to be updated. This problem led to the investigation into other forms of navigation that would be possible once a spacecraft leaves the solar system, such as positioning based on observing interstellar wind [45] and interstellar magnetic fields [46]. However, the most promising navigation alternative seemed to be pulsars.

Initial investigation on the topic of pulsar based navigation was stimulated by the question of assessing how useful and how accurate it can be. Of particular interest was the issue of the ambiguity problem. It will be covered in great detail in the following sections, but the first investigations of the problem involved modeling the ambiguity problem in one dimension and seeing how far out in space or time an observer must travel before a signal composed of multiple pulsar waves repeats itself. But it was not initially clear which topic in pulsar navigation should be investigated because there was little support for research on the topic at Texas A&M. Nobody in the department worked on a similar kind of navigation problem, and the physics department didn't have anyone involved in pulsar astronomy. Though there was ongoing work on X-Nav going on at NASA Goddard, establishing a working relationship with the center was difficult, and thus, in order to make up for this deficit of support The Aggie Research Program (ARP) at Texas A&M was leveraged. The ARP provided a way of connected graduate students with undergraduate students interested in gaining research experience. After an initial recruiting phase, a small team of undergraduate students was assembled to help explore various topics related to the field of pulsar astronomy and pulsar based navigation. Various aspects of signal processing, position estimation, statistical analysis, mission simulation, GPS signal analysis, and ambiguity resolution were studied. After exploring the topics it seemed that the question of ambiguity resolution offered an interesting problem that was rich enough to provide work for research.

The team's work on the ambiguity problem led towards the proposal of the Spheres Algorithm which was initially developed by Samuel McConnell and Michael McCarthy and was presented at the 2018 AAS/AIAA Astrodynamics Specialist Conference in Snowbird, Utah [47]. For the students involved it their opportunity to be directly involved with academic research efforts, as well as their first opportunity to travel to an academic conference, let alone present. Spheres was a partitioning algorithm which split up a spatial search domain in order to eliminate impossible candidate solutions from intersections of pulsar signals. Methods of optimal packing and partitioning were investigated. Ray tracing algorithms and rasterizing methods were also studied as ways of searching through a discretized domain and selecting relevant partitions. This algorithm served as the foundation for the ambiguity resolution algorithm presented in the following chapter, and while Spheres has been largely reworked, and little remains of the original concept, it is important to note that the work started from ideas developed by undergraduate researchers.

Background on the X-Nav Position Estimation Problem

The key behind using pulsar signals to help estimate position is the ability to measure precisely the arrival time of an observed pulse and compare that time to an expected arrival time at a predetermined reference location. Hence, *X-Nav provides a position estimate not by locating an observer with respect to the pulsars themselves, but by locating an observer with respect to a chosen reference point.* Astronomers have observed different pulsars from various locations on Earth, and to coordinate their efforts, the agreed upon standard reference location at which expected arrival times are cataloged is the Solar System Barycenter (SSB), which oscillates around the center of the Sun, sometimes being inside the Sun and sometimes outside. Since there is no actual observer or receiver at the SSB, all cataloged arrival times are mathematically translated and converted to show when an observed pulse either *would* arrive or *would have* arrived at the SSB. Thus, an implementation of X-Nav using the currently available pulsar time data would produce position estimates with respect to the SSB. Other reference points can be chosen, and all the arrival times can be recomputed, however for the contents of this dissertation the SSB is considered the reference point that all arrival times are cataloged with respect to.

A second key behind X-Nav is the assumption that *the signal coming from a pulsar can be modeled as a planar wavefront.* This means that all points lying on a plane perpendicular to the direction of pulse propagation see the pulse arrive simultaneously. If one were close to a pulsar, such an assertion would not be possible and the path described by the rotating beam would produce a spiral radiating outward producing spherical wavefront. However, given that the closest millisecond pulsar, PSR

J04374715, is over five hundred light years away it is reasonable to assume planar wavefronts when considering the problem of position estimation around Earth.

Phase of Wavefronts

When considering the line or plane representing a wavefront it is typically associated with the peak of the signal which is assigned a zero phase. In the following sections, this standard is assumed when attempting to determine position of an observer with respect to a reference point. Later on, there will be examples of an observer observing multiple peaks simultaneously at a single point in space. Realistically, such a scenario is highly improbable, however the choice of phase associated with a wavefront is arbitrary. Wavefronts can be aligned at their pulse peaks or any other part of the signal. Therefore, there is no loss of generality in assuming that wavefronts of observed pulsars can converge on a single point in space and time, since those wavefronts can be assigned to any phase of the signal, not only the peaks.

Single Pulse X-Nav

To illustrate how arrival times are used to estimate position, we consider a simplified scenario where a pulsar emits only one single pulse. This is shown in Figure V.1 for an observer located at position $\mathbf{p}$ relative to a reference point and a single pulse traveling in direction $\mathbf{n}$.

The pulse moves at the speed of light, c, and the difference in time between the pulse passing the reference point and the pulse arriving at the observing spacecraft is called the ΔTOA, where TOA stands for Time Of Arrival. Realistically, mea-

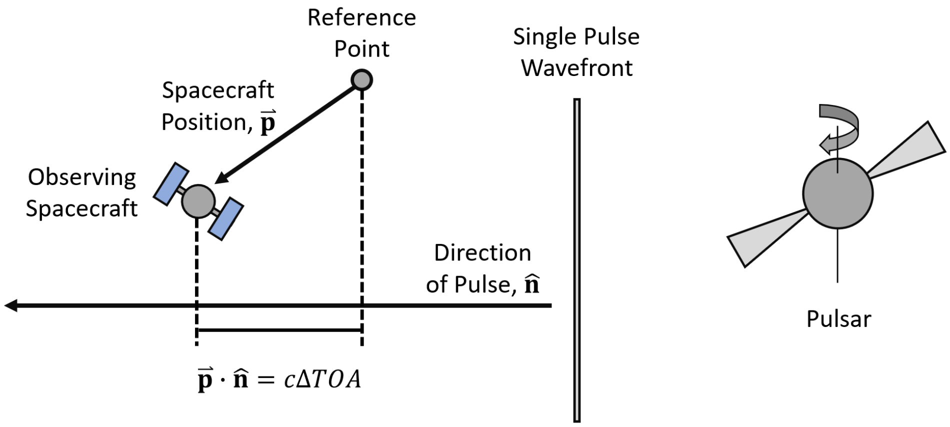

Reference Point
Single Pulse Wavefront
Spacecraft Position, $\vec{\mathbf{p}}$
Observing Spacecraft
Direction of Pulse, $\hat{\mathbf{n}}$
Pulsar
$\vec{\mathbf{p}} \cdot \hat{\mathbf{n}} = c\Delta TOA$

Figure V.1: Simplified diagram showing relationship between observing spacecraft, reference point, and incoming pulse.

sured TOAs are not perfect and are affected by timing noise due to inaccuracies in measuring exactly where the peak of the signal lies. This timing noise will be incorporated later, however for the purposes of this example it is ignored. The observer is assumed to have a catalog of expected arrival times for pulses at the reference point. By looking at the difference in arrival time between when the pulse was seen and when it was expected, it's possible to calculate the distance from the observer to the reference point along the direction of the pulse propagation.

One pulsar is not enough to give an observer a position fix in three dimensions, however if three pulses are observed with three ΔTOA values coming from three distinct directions, $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$, then the system shown in Equation 5.1 can be formulated.

$$
\begin{aligned}
\mathbf{n}_1 \cdot \mathbf{p} &= c\Delta TOA_1 \\
\mathbf{n}_2 \cdot \mathbf{p} &= c\Delta TOA_2 \\
\mathbf{n}_3 \cdot \mathbf{p} &= c\Delta TOA_3
\end{aligned}
\tag{5.1}
$$

This system can be rewritten as the matrix equation

$$
\begin{bmatrix} \mathbf{n}_1^{\mathrm{T}} \\ \mathbf{n}_2^{\mathrm{T}} \\ \mathbf{n}_3^{\mathrm{T}} \end{bmatrix} \mathbf{p} = c \begin{bmatrix} \Delta TOA_1 \\ \Delta TOA_2 \\ \Delta TOA_3 \end{bmatrix}
\tag{5.2}
$$

where the matrix on the left-hand side containing the directions of the pulses is 3×3 and is invertible as long as the directions are distinct. Thus, Equation 5.2 offers an easy and direct way to solve for the position of the observing spacecraft using observations from three different pulsars.

Single Pulse X-Nav with Clock Drift

Over a long duration mission, an observer's clock can drift with respect to reference time if there are no synchronization updates. If this happens, then the measured TOA of pulses will be offset. Fortunately, X-Nav can work in this scenario as well, and not only can it still estimate the observer position, but it can also estimate the clock drift. To illustrate how this works we assume that we have an observer at position **p** and a pulse traveling in direction **n**. We then formulate a new relationship for determining ΔTOA that includes clock drift:

$$
\begin{aligned}
TOA_{reference} &= TOA_{observer} + \Delta t + \delta_\tau \\
\Delta TOA &= \Delta t + \delta_\tau
\end{aligned}
\tag{5.3}
$$

where δ is the clock drift and where Δt represents the time it takes for a pulse to traverse the distance $\mathbf{p} \cdot \mathbf{n}$:

$$\Delta t = \frac{\mathbf{p} \cdot \mathbf{n}}{c} \tag{5.4}$$

If a clock is perfectly synchronized and there is no clock drift with respect to reference time, then $\delta_\tau = 0$ and $\Delta t = \Delta TOA$. If we assume that four distinct pulsars are observed then combining Equations 5.3 and 5.4 we can create the system

$$\begin{aligned}
\mathbf{n}_1 \cdot \mathbf{p} + c\delta_\tau &= c\Delta TOA_1 \\
\mathbf{n}_2 \cdot \mathbf{p} + c\delta_\tau &= c\Delta TOA_2 \\
\mathbf{n}_3 \cdot \mathbf{p} + c\delta_\tau &= c\Delta TOA_3 \\
\mathbf{n}_4 \cdot \mathbf{p} + c\delta_\tau &= c\Delta TOA_4
\end{aligned}$$

which can be written in matrix form

$$\begin{bmatrix} \mathbf{n}_1^{\mathrm{T}} & 1 \\ \mathbf{n}_2^{\mathrm{T}} & 1 \\ \mathbf{n}_3^{\mathrm{T}} & 1 \\ \mathbf{n}_4^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ c\delta_\tau \end{bmatrix} = c \begin{bmatrix} \Delta TOA_1 \\ \Delta TOA_2 \\ \Delta TOA_3 \\ \Delta TOA_4 \end{bmatrix} \tag{5.5}$$

The matrix on the left hand side is a 4×4 matrix which is invertible if the pulse directions $\mathbf{n}$ are distinct. Multiplying Equation 5.5 by the its inverse gives a solution for $\mathbf{p} = [x, y, z]^{\mathrm{T}}$ and δ_τ. This formulation can be extended to more than four pulsars, however the solution of Equation 5.5 becomes a least-squares solution. This will be explored more in later sections.

Taking Advantage of Pulse Stability to Pre-Compute Arrival Times at Reference

The stability of the pulsar signals means the arrival times are very predictable. This means that the arrival times at a specific reference point can be all be pre-computed for the foreseeable future. This can be done for multiple different reference points as well. Thus, it is plausible that a spacecraft employing X-Nav would have an on-board catalog of arrival times for all observable pulsars at various pre-determined reference points around the solar system. If the reference points are placed at relatively frequent intervals throughout the solar system, or any navigational space of interest, then the planarity assumption can always be made, as long a spacecraft is relatively close (tens of millions of kilometers) to the reference point.

Scope of X-Nav Position Estimation Problem and List of Assumptions

To demonstrate the full set of potential capabilities as well as limitations of an X-Nav system very high fidelity models of light propagation through interstellar medium, stellar astrophysics, and signal processing are needed. In addition, in order to ensure that all potential operational modes are considered, spacecraft traveling at non-constant velocities, rotating, and performing maneuvers would need to be modeled, in addition to modeling the physics and operations of the detecting array, antenna, or imager. This level of fidelity stretches far beyond the scope of this dissertation. Therefore, to help contextualize the work and help clarify its applicability, consider the following list of assumptions being made when addressing the problem of X-ray pulsar-based position estimation.

Assumptions Behind X-Nav in this Dissertation

The problem of X-Nav is complex and can feature many different physical processes, and thus it is important to constrain the analysis by making some simplifying assumptions.

Relativistic Effects Ignored: Large gravitational bodies, such as gas giants like Jupiter, or other distant stars that lie near the path of incoming light from a pulsar could deflect incoming pulses and change their direction and time of arrival. Gravitational lensing like this is ignored and directions of pulse propagation and signal periods are considered constant.

Fixed Reference Point: The location of the reference point is considered fixed in inertial space with respect to observed pulsars. Additional reference points may be defined, but they will each also be fixed with respect to the celestial reference.

Constant Spacecraft Velocity: Observing spacecraft velocities are known, constant, and far below the speed of light. If a spacecraft has a non-zero velocity with respect to the reference point, this velocity is known and constant. Orbital mechanics and and thrusting maneuvers are not considered.

Simple Signal Processing: The pulse signal data acquisition process and signal processing are not modeled. The complications of photon capture, epoch folding, channel alignment, gathering enough photons to be statistically significant and be able to model the incoming pulsar signal are not considered. It is simply assumed that the spacecraft observes incoming pulses and is able to associate an arrival time for the peak of each pulse and that *TOA* measures have timing noise which follow a zero-mean Gaussian distribution with known standard deviation. Signal periods and phase are assumed to also be known.

Planar Wavefronts: Wavefronts are assumed planar. Considering the closest millisecond pulsar is over five hundred light years this assumption is appropriate for navigating distances on the order of tens of millions of kilometers. If a larger navigational space than that is being considered, then it is either necessary to use multiple reference points or incur errors if the planarity assumption is not relaxed. Another alternative is to ignore signals from all pulsars which are too close for the assumption to be valid. This assumption also, along with the assumption that periods are constant, implies that distances between wavefronts in a pulse chain are uniform along the entire wavefront.

Bounded Navigational Space: The navigational space being explored is on the order of tens of millions of km. Exploring a space outside of that range will begin to interfere with the planar wavefront assumptions. The tighter the timing noise, and the more precisely *TOA*s are known, the smaller the volume of space for which the planarity assumption is valid.

Constant Periods: Pulsar periods are assumed constant and known with perfect accuracy. The period of a pulsar its tied to its angular momentum and rotational kinetic energy, both of which are very stable. There are some small sources of variation in the angular spin rate of a pulsar. Pulsars slowly lose mass, and thus angular momentum, as they burn and radiate out energy. Pulsars which have slight asymmetries on their surface or that are part of a binary system can also bleed kinetic energy through the radiation of gravitational waves. It's also possible that pulsar are surrounded by an accretion disk and they are actively sucking up matter, which would cause their spin rate to increase. Finally, pulsars can "glitch", in which their rotational frequency might suddenly shift by a factor of about one millionth due to small changes in internal processes of the star or starquakes. Aside from the glitches, most other changes in period are slow and predictable, so even when periods are not constant their variation can be accounted for. These variations, however will not be considered in this work.

Interstellar Medium Ignored: The effects of interstellar medium on pulsar signals are not considered. Dispersion between light at different wavelengths is ignored. Any refraction or change in direction of pulse propagation is either not considered or is considered constant and applies equally to all wavefronts.

Background on the Ambiguity Problem and Multi-Pulse X-Nav

The ambiguity problem is a data association problem which occurs when an observer is lost in space and/or time and has poor *a priori* estimates of their position and clock. While the regularity of pulsar signals provides a means of accurately predicting arrival times, the lack of variation or modulation between successive pulses in a chain makes it difficult to discern one pulse from the next. Thus, when an observer marks the TOA of a pulse and they want to compare it to $TOA_{reference}$ there are actually multiple possible candidates for comparison. *This is the ambiguity problem: "How can an observer tell which pulse TOA in a chain of TOAs they have observed in order to calculate the correct ΔTOA?"* Given

$$\Delta TOA_k = TOA_{reference,j} - TOA_{observer,i}$$

how can we ensure that $i = j = k$? Failing to correctly associate the pulses and being off by even just one period can shift an observer position estimate by thousands or even tens of thousands of kilometers for slower-rotating pulsars. The degree of difficulty of resolving the ambiguity problem depends on the confidence of *a priori* time and position estimates. If one is very confident in their clock and position estimate, then the problem is easy, though then there is less utility in using the method in the first place. The problem is visualized in Figure V.2. Assume that a signal, traveling at the speed of light c, with wavelength λ is observed by a spacecraft with position uncertainty δ_r and timing uncertainty δ_τ from clock drift. Realistically, position and time uncertainty would be modeled by Gaussian distributions, however for the purposes of this section, let it be assumed that position and time uncertainty are distributed uniformly on the range $[0, \delta]$. For the given scenario, there needs to be a way of determining if there is risk of an ambiguous solution or not. If

$$\lambda - c\delta_\tau > \delta_r$$

then there is no ambiguity. This is represented by the black circle in the Figure V.2. The previous estimates of position and time are precise enough, i.e. the observing spacecraft is confident enough in them, that it is obvious which pulse was observed and there is no ambiguity. However, if

$$\lambda - c\delta_t < \delta_r$$

then there is too much uncertainty to know for sure which pulse was seen. This is represented by the red circle in Figure V.2. As a least upper bound of the conditions which create ambiguity, if $\delta_\tau \geq T$, where $T = \lambda/c$ and is the signal period or if $\delta_r \geq \lambda$ then there is guaranteed to be ambiguity in the observation.

As an example, consider a spacecraft with position uncertainty of $\delta_r = 1000km$ and time uncertainty $\delta_\tau = 0.0005s$ observing a pulse chain from a millisecond pulsar with period $P = 0.01s$ and wavelength $\lambda = 3000km$ moving at $c = 3\text{e}5km/s$.

$$\lambda - c\delta_\tau > \delta_r$$

$$3000km - 150km > 1000km$$

In this scenario, there is no ambiguity. However, if the period/wavelength is shortened, or the timing or spatial uncertainty increased, that may change.

If an observer happens to be dealing with the scenario of potentially ambiguous observations, then there are two main approaches to resolving this problem:

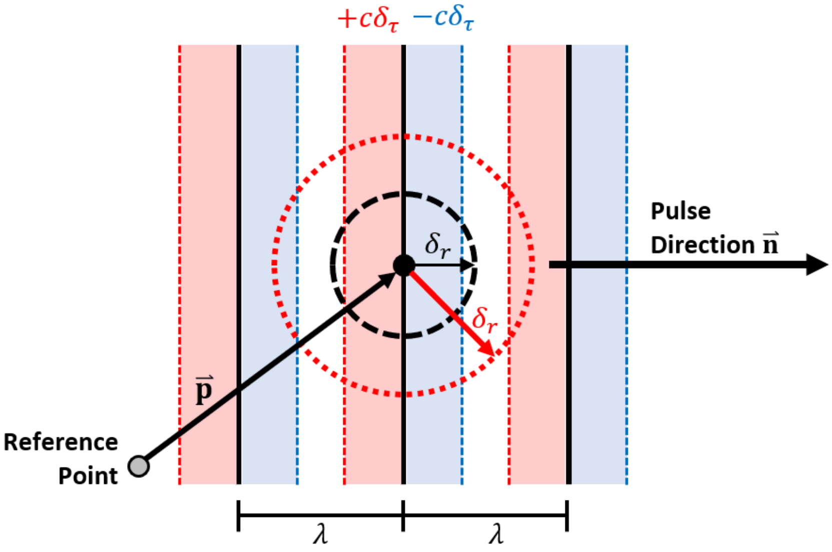


Figure V.2: A spacecraft with time and position uncertainty observing pulses from a pulsar. For the given timing uncertainty δ_τ the black circle represents a non-ambiguous measurement scenario, while the red circle represents an ambiguous measurement scenario

1. Improve position or time estimate using other measurements and state estimation tools. If uncertainty in the position estimate can be reduced far below that of the wavelength of the pulse signals and the timing uncertainty reduced far below that signal period, then there is no ambiguity and the association problem is easy to solve.

2. Just like superimposing waves create wave groups, so too can collections of repeated measurements from multiple different pulsars create measurement groups. Generally, the more waves of different phases and frequencies are brought together, the lower the fundamental frequency of the overall signal and the larger the spacing between wave groups. Similarly, by observing more

pulsars, an observer can create local snapshots of the pulse wave landscape which do not repeat until one either waits a very long time or moves a great distance. Thus, the effective period and wavelength of the observed pattern is increased, which allows position and time drift estimation even when *a priori* knowledge of either might be poor.

Implementing the second approach boosts the capabilities of X-Nav and makes it a more capable navigation scheme. Perhaps it's possible with enough observed pulsars that an observer can be truly lost in space, with position uncertainty of millions of kilometers and clock drift on the order of days or weeks, and still be able to find out where and when they are. The more pulsars they can confidently identify and process their signals, the higher the chances of them finding a unique navigation solution. *Thus the ambiguity problem can be reformulated as: Given a particular set of pulsar signals $[P_1, P_2, ..., P_N]$ each with known period, how lost, in space and time, can an observer be and still successfully implement X-Nav to estimate their position and clock drift without worrying about ambiguous observations?* To begin to answer this question, we will first simplify the problem and look at it separately in one and two dimensions.

Ambiguity in One Temporal Dimension with Multiple Pulsars

Consider first the one dimensional example of having a static observer who is measuring pulse arrival times from three different pulsars with periods $T_1 > T_2 > T_3$ as shown in Figure V.3. Suppose this observer notices an alignment of the pulses so that all three happen to arrive at exactly the same time. How long does the observer have to wait until they observe the same pattern? For the chosen periods, after 3 periods of T_1 the same alignment is observed. Thus at intervals of $3T_1$ the observer will see the same pattern repeat. This is the minimal time between repeating solution and will be denoted as T_{Min}. If the timing uncertainty is δ_τ is greater than $3T_1$ then the observer can't be certain about which alignment they observed, and thus they have at least two possible times associated with their observation.

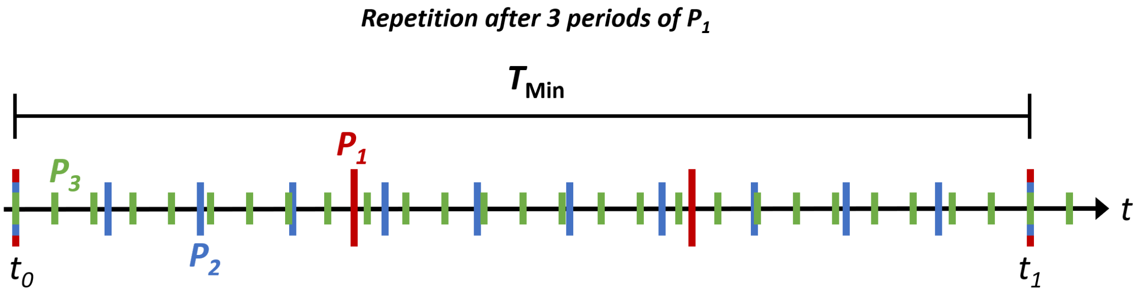


Figure V.3: Observation times of pulse chains coming from three different pulsars with periods $T_1 > T_2 > T_3$. The alignment of all three pulses repeats after 3 periods of T_1

To get an idea of how accurate our clock needs to be, we want to know what is the shortest possible time interval between t_0 and t_1. If the pulsar periods are represented as integers, then this is the well known problem of finding the least common multiple (LCM) between integers that has been studied for thousands of years and it is solved by implementing the Euclidean Algorithm to first find the greatest common divisor first. Even if the arrival times are not initially recorded as integer values, it is possible to scale the units of the arrival times until they are represented as integer multiples of very small time units and then find their LCM.

Thus, the one dimensional problem considering time ambiguity is readily solved.

Ambiguity in Two Spatial Dimensions with Multiple Pulsars

Consider an observer who makes an instantaneous observations, and again sees three simultaneous pulses coming from three different pulsars. The two dimensional representation of such an arrangement is show in Figure V.4. Note that as more pulsars are observed the distance between ambiguous solutions grows.

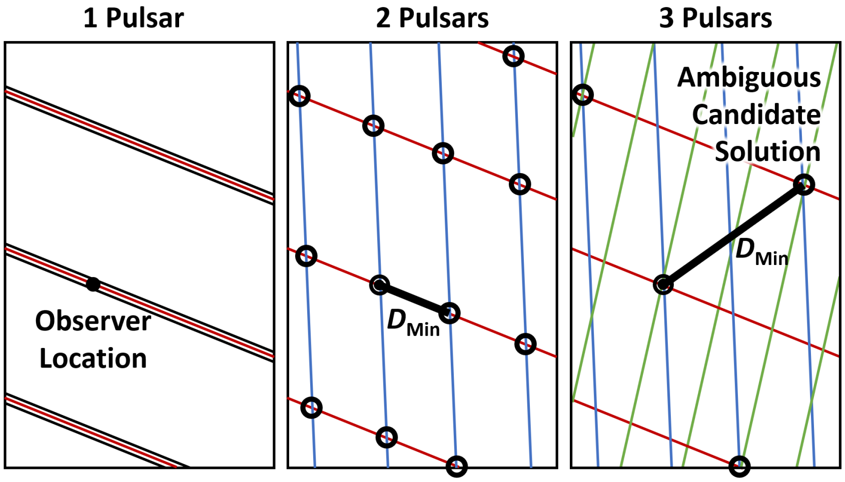


Figure V.4: A 2 dimensional view of multiple pulses coming from pulsars. **Left:** An observer could be anywhere on any wavefront when considering only one pulsar. **Middle:** An observer sees pulses from two distinct pulsars. They can only observe both pulses simultaneously at the intersection points. The minimal distance between possible solutions is D_{Min}. **Right:** As the number of observed pulsars increases the distance between candidate solutions grows.

If D_{Min} is greater than the position uncertainty δ_r than an observer can confidently identify which solution is being observed. The value of D_{Min} depends on the number of observed pulsars, their periods, and their directions of propagation. The intersecting wavefronts create a set of lattice points and there exist algorithms to find the minimal distances between these lattice points.

Ambiguity in Two Spatial Dimensions and One Time Dimension with Multiple Pulsars

We extend our analysis to a finite time and spatial domain. Consider again the two dimensional space as before, however now the wavefronts move over time. Referencing Figure V.5 suppose that a solution is observed at the observer location at t_0. A short time later at t_1 a different solution is observed nearby.

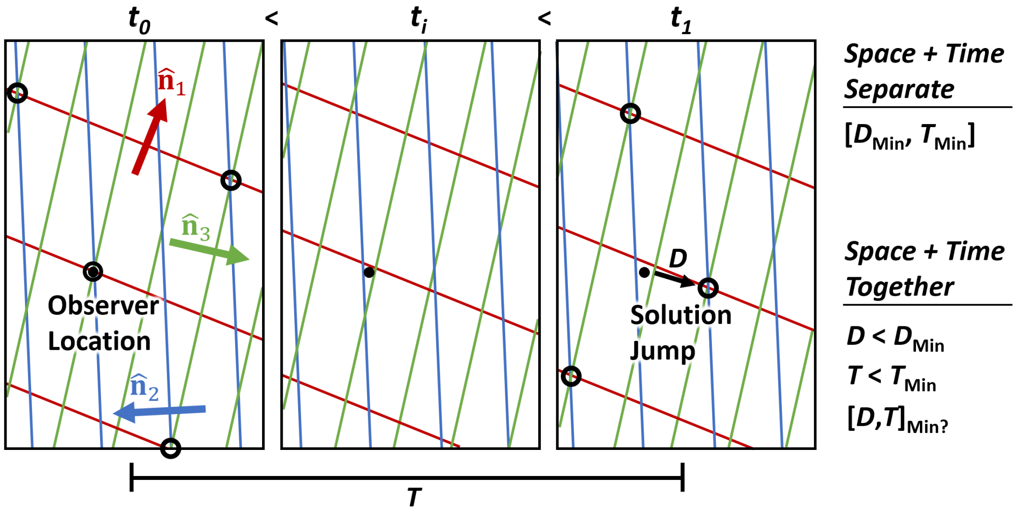


Figure V.5: A two dimensional space of pulses at different times. At t_0 a solution is observed at the observer location. At time t_1 the nearest solution jumps and is no longer where the observer is.

Previously, when considering only the two dimensional case, with no variation in time, D_{Min} was the shortest distance between solutions. Now the distance between solutions needs to be measured in both time and space. One possibility is to use a Minkowski space-time formulation using either the complex form $\mathbf{u} = (x, y, ict)$, where c is the speed of light and i is the imaginary unit, or real form $\mathbf{u} = (x, y, t)$. The norm for such a vector is defined as $\|\mathbf{u}\| = \sqrt{c^2t^2 - x^2 - y^2}$. The utility of such a formulation has not been investigated, however. As of the time of this writing it is not known to the author if there exists a minimal space-time metric which separates candidate solutions.

The key fact to understand is that the minimal distance between solutions can now be less than than D_{min} and the minimal time between them be less than T_{Min}. The implications of this are that even if for a given instant in time D_{Min} is greater than δ_r and at a given point in space T_{Min} is greater than δ_τ, this is no longer enough to assure that there are no ambiguous solutions. It's possible that with a very small time increment and small displacement, both of which are within the bounds of δ_τ and δ_r that an ambiguous solution may exist. Thus a new minimal time step or spatial step must be used. This will be discussed in a future chapter.

Extension of the ambiguity problem into three spatial dimensions is not investigated in this dissertation. However, with the assumption of planar wavefront, the concepts developed in these examples with two spatial dimensions and one temporal dimensions apply to any number of spatial dimensions.

The Ambiguity Problem with Timing Noise

In order to fully formulate the ambiguity problem, the inclusion of pulse timing noise is required. Pulse timing noise is not to be confused with timing uncertainty δ_τ associated with clock drift. Pulse timing noise is the uncertainty that comes with trying to pinpoint at exactly what point in time a pulse is observed. As previously mentioned, pulses actually take the form of a chain of photons which exhibit shot noise — arriving according to a non-homogeneous Poisson arrival process — traveling along different decoherent frequency channels. Timing the exact arrival of a pulse is not a deterministic process but a stochastic one which requires the use of epoch folding in order to gather enough statistics about a signal to reproduce the waveform. While clock drift δ_τ affects the timing of all pulse signals from all pulsars equally, the timing noise, though it might follow a fixed distribution, will produce different variations in the timing of observed pulses. An example of timing noise can be seen in the work of Franzen [1] where several hours of photon collection were used to reconstruct a signal from the Crab Pulsar. After the signal was processed, the arrival phase of the peak was determined to within ±0.0004 parts, which for a signal with a period of about $33.5ms$, translates to $\pm 13.4\mu s$ of timing noise. For a pulse traveling at $c = 3\text{e}5 km/s$, this gives an uncertainty in the location of the pulse wavefront of about $\pm 4km$ at an instant in time. The exact value of the timing noise is not important, however it will be assumed that it is three to four orders of magnitude smaller than the period of the pulse signal.

Timing Noise – Revisiting the 1D Temporal Problem

We now re-examine the one dimensional ambiguity problem while also introducing timing noise σ_t. Again we are interested in finding the shortest time interval between candidate solution observations, as that represents the most uncertainty in clock drift that one can have before they must worry about ambiguous solutions. Referring to Figure V.6 we note that again at t_0 we have alignment of all pulses in time. Also, as before, there are three pulse signals with periods $T_1 > T_2 > T_3$ which are known very precisely however the periods have been scaled so that they do not perfectly line up in time after three periods of T_1 as was the case in Figure V.3. Also, in this scenario the arrival times can only be measured with a resolution of $\pm\sigma_t$. This means that if the estimated TOA of two pulses is separated by less than $2\sigma_t$ they are considered to be simultaneous events.

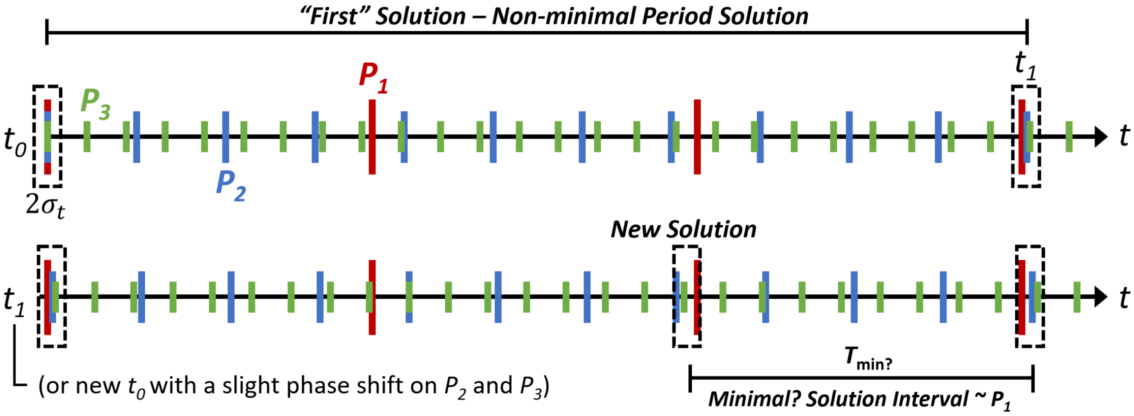

"First" Solution – Non-minimal Period Solution
t_1
P_1
P_3
t_0
t
P_2
$2\sigma_t$
New Solution
t_1
t
$T_{min?}$
(or new t_0 with a slight phase shift on P_2 and P_3)
Minimal? Solution Interval ~ P_1

Figure V.6: The one dimensional temporal ambiguity problem revisited with timing noise added. The minimal time between solutions T_{min} is about T_1

Again, realistically, timing noise would be modeled likely by a Gaussian distribution meaning that two pulses which have been observed farther apart, but still separated by less than $2\sigma_t$ have a lower probability of having arrived simultaneously than two pulses which have arrive close together. However, for this example we consider a uniform distribution of noise over the range $2\sigma_t$, and thus the classification of simultaneity between pulses is binary: if their spacing is less than $2\sigma_t$ they are considered simultaneous, if it is greater then they are not simultaneous. A point in time where all three pulses are observed simultaneously represents an ambiguous candidate solution.

It's important to note that as the time is progressed the first candidate solution encountered at t_1 is roughly three periods of T_1 from the first solution at t_0. However, this is not the minimal amount of time between two possible solutions. If one continues to progress time past t_1, or simply restarts the clock at t_0 with a slight shift in phase for T_2 and T_3 it's possible to encounter a shorter spacing between solutions. In this case, it can be seen that the shortest interval between candidate solutions, T_{min} encountered in the given time span is about T_1. The exact length of the solution interval depends on how the solution windows are assigned a TOA with respect to the pulses and how the gap between solutions is measured. If we adopt a scheme where a solution is assigned a TOA which is equidistant from the two farthest pulses in the solution window, then the shortest interval between solution window centers will be T_1 and this would only occur if somehow all the shorter period solutions fell exactly on the pulses of T_1 which is not possible for the configuration in Figure V.6.

One of the difficulties created by the inclusion of timing noise is that even if the periods could be written as integer values of a very small time unit, the Euclidean algorithm could no longer be implemented. In addition, the time to the first encountered solution after some initial time t_0 is not necessarily the minimal possible time between solutions. Therefore, simply running the clock forward and measuring intervals between observed solutions is not an effective method of estimating $T_{\min}$, as it is not clear when one has found the absolute minimal gap between solutions. Thus, a different method is needed for finding this minimal time between solutions and it is discussed in the next chapter.

Timing Noise – Revisiting the 2D Spatial Problem

Extending the effects of timing noise into the two dimensional domain, we first look at a near intersection of single pulses from three different pulsars. We do this first by giving a finite width to each line representing a wavefront. While previously the lines where infinitely thin, they now have a finite width of $2c\sigma_t$. Referencing Figure V.7 we can see that when three pulses of finite width intersect, or nearly intersect, their areas overlap to produce a convex polygon. When dealing with only three pulsars these shapes can vary from triangle, to quadrilateral, to pentagon, and finally hexagon, however the shape the overlapping area produces is always convex.

Like in the one dimensional case, the inclusion of timing noise increases the number of potential solutions and decreases the intervals between ambiguous solution candidates. Looking at the two dimensional problem with multiple pulses from three pulsars shown in Figure V.8 there are multiple solution candidates even though there is only one exact solution. The periods are assumed to be known with perfect precision, so on average the gap between successive pulses remains constant, even

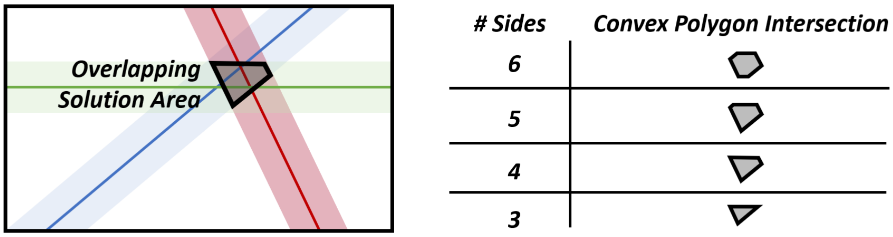


Figure V.7: Intersecting bands from three pulsars. Overlapping shape is convex polygon with three to six sides.

though there is uncertainty regarding when exactly each pulse arrives.

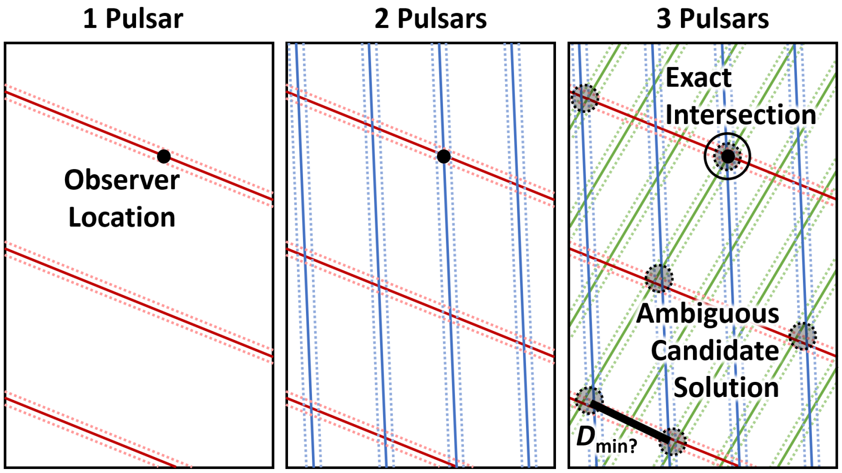


Figure V.8: Candidate ambiguous solutions for multiple pulses with timing noise from three pulsars in two dimensions. Note that only one exact solution exists in the observed space where all the pulses line up perfectly.

The figure was generated by taking Figure V.4 which demonstrated repeating solutions with exact intersection in two dimensions and altering the short period T_3 pulses to have a shorter period and different direction of propagation. For this new configuration the distance between solutions can be measured by either taking the distance between the centers of the solution polygons or by finding the two points on two separate solution polygons that are closest to one another. In either case, it is clear that the minimal distance between solutions is much shorter than what it would be if there was no timing noise. The shortest distance between two candidate solutions is denoted by $D_{\min}$ in the figure. Note that there happens to be an exact intersection in the space being considered, however it is the only one, so any other exact solutions which might exist are significantly farther awar than the $D_{\min}$ between the two ambiguous candidate solutions.

Timing Noise – 2 Spatial Dimensions and 1 Temporal Dimension

The full complexity of the ambiguity problem being considered can be seen once pulses are allowed to shift over time. Relative to the one and two dimensional cases, and the cases without timing noise, this case will allow for significantly more candidate solutions. For a sufficiently large enough search space and long enough period of time these candidate solutions might be popping in and out of existence by the thousands at every time step. To resolve the ambiguity problem, not only must one find out what the shortest possible time and distance is between solutions but one must also try to produce a position estimate by eliminating false candidate solutions. The Pulse Ambiguity Resolver Algorithm which achieves this is presented in the following chapter.

Setting the Stage – State of the Art, Literature Review, and Available Frameworks

Before diving into algorithm development, it is worth noting what resources and frameworks are available for tackling the ambiguity problem. The following sections contain information regarding new developments and improvements in the field of X-Nav as well references and explanations to some of the foundational work and tools available for continuing work in the field.

Literature Review

X-ray Pulsar Navigation Algorithms and Testbed for SEXTANT: Winternitz et. al present describe the design of the SEXTANT technology demonstration as well as the design and capabilities of the GXLT testing platform. [48]. This platform represents some of the latest developments in X-Nav being worked of at Goddard Space Flight Center. The architecture is capable of simulating the generation of pulsar signals as well as signal processing.

Introduction to X-Nav and Feasibility Studies: Becker's work [49] provides an introduction to the basic techniques behind X-Nav as well as clear descriptions and visualizations of the ambiguity problem. The ESA's feasibility study [50], covers in detail many aspects regarding hardware, algorithms and the physics of signal generation and processing and provides clear steps for integrating the results of X-Nav into navigational filters.

Spacecraft Navigation Using X-Ray Pulsars: Sheikh et. al present a thorough summary of the developments in the field of X-Nav as well as some of the challenges that must be overcome. [51]. First it's pointed out that newly born neutron stars rotate with periods on the order of tens of milliseconds. Older stars eventually slow down to the order a several seconds. This rotation is very stable and predictable. Accretion X-ray pulsars are divided into those with a high mass binary companion and a low mass binary companion. X-ray signal stability is greater than radio because of reduction in propagation effects from the interstellar medium on X-ray light. The authors have cataloged 140 X-ray pulsars and Parkers Multibeam Pulsar Survey database has listed many pulsar characteristics. Standard profile templates are created by observing pulsars over extended periods of time and averaging the period or "epoch folding". This produces a signal with a very high SNR. Radio pulsars require huge receivers and broadband radio sources including local celestial galaxies add noise. X-ray objects are very dim, for example the Crab Pulsar is at 9.9×10^{-9}erg/cm^2/s. Inaccuracy in position knowledge of pulsars allows the use only of their direction, while distance to the pulsar cannot be used. Few sources are available out of the galactic plane, leading to problems in finding signals coming from unique directions. A priori estimates of pulse profiles or some other method to identify correctly which specific pulse is detected is necessary. Some pulsars, such as most accretion X-ray pulsars, exhibit glitches, which are changes in their spin rates. These glitches make them unreliable for navigation purposes. A figure of merit is introduced to assist in identifying X-ray sources with potential to provide good timing and range accuracy. It is theoretically possible to achieve accuracies of about 100m with 100s observation windows and a 5 m^2 antenna.

Time Corrections for Offset from Barycenter: The time offset must account for relativistic effects for a clock in motion about a gravitational body. It must convert satellite clock to inertial coordinate time (TDB) clock. For distant pulsars direction of pulses can be considered a constant for any movement within the solar system. For accuracy under 300m relativistic time correction must be included which includes gravitational effects of all large planets in the solar system.

Pulsar Based Nav: A full navigational system would include: a sensor to detect the pulse, a clock on board, and a database of known timing models for the pulsars. Static detectors would collect signals coming into FOV and compare them to a catalog, while for moving sensors, a gimbaled system would perform faster due to its ability to track targets. Doppler shift can be used to determine vehicle velocity and pulsar signals can be used to stabilize atomic clocks. This can be done by implementing a phase-locked loop. *TOA*s can also be used to correct clock error based on how far off the on-board clock times are the incoming pulse. A Kalman filter (KF) can be made to estimate the bias scale factor and jitter of the clock.

Pulsar Elevation Method: 4 methods presented. 1st and 2nd method are comparable to OpNav techniques, however are more robust to blinding of detectors/ blocking of light. The two methods that use accurate pulse *TOA* measurements require the coordinate time conversion and the barycenter offset corrections to be applied. This requires that the spacecraft know its positions in order to compute these corrections, so there is a chicken and egg problem. On-board attitude is known and used to get direction towards pulsar. Simultaneous observation of planetary body, elevation angles between source and reference body, and apparent size of reference body can be used to provide position relative to body.

Delta-Correction Method: Use an on-board position estimate and improve it by measuring phase discrepancy. Combining measurements from multiple pulsars solves the offset problem in three dimensions. Knowing the correct phase shifts requires knowing spacecraft position relative to solar system bodies in order to make relativistic corrections. Any significant vehicle motion occurring during the time space between measured and predicted pulse arrivals must be taken into consideration. To measure performance the time equations can be linearized. There will be geometric dilution of precision (GDOP) which is based on the covariance matrix of the estimated errors of the position estimate. A more even pulsar distributions will produce less GDOP. Measurements are uncorrelated so the covariance matrix can be expressed as diagonal. The Unconventional Stellar Aspect (USA) Experiment [52], which had 1000 cm^2 effective area, and it's results can be used to evaluate peroformance of X-Nav systems. Comparing experimental results to USA results and using template files to get observation TOA's, it's possible to get position offsetson the order of ± 10 km.

Studies of Ambiguity Resolution: Several studies have been performed which directly look at the ambiguity problem and propose ways of resolving it. Huang [53] offers methods built on hypothesis testing and an SVD representation of the acceptance space that builds upon previous approaches at solving the problem.

Li [54] proposes a least squares and ambiguity covariance approach and Feng [55] investigates the phase ambiguity problem while also addressing the problem of GDOP.

Theoretical and Conceptual Frameworks

From an algorithmic perspective, the problem of X-Nav has many components that leverage tools from many different fields of mathematics, physics, and engineering. However, the focus of the developments presented in this dissertation is on resolution of ambiguity when processing multiple pulses from different pulsars, thus, theory related to pulsar astrophysics, signal processing, and stochastic processes is not heavily leveraged. The frameworks leveraged heavily in this work are listed below.

Parametric and Implicit Representations: Resolving the ambiguity problem requires heavy use of both parametric and implicit descriptions of geometries and movements of points. Pulse wave fronts are represented both in parametric form using normal vectors and propagation speed and in an implicit form when studying intersections between signals.

Least Square Techniques: Least square techniques are used for estimating solutions to the multi-line intersection problem and attempting to find points in space and time which best correspond the observed pulsar signals.

Euclid's Algorithm and LCM: For analyzing a simplified, one dimensional version of the ambiguity problem, pulsar signal periods can be represented using integers and thus Euclid's algorithm can be used to look for repeating patterns in the signals. This approach does not work well once the problem is extended into multiple dimensions.

Fundamentals of Geometry: Resolving the ambiguity problem involves representing potential intersection areas and volumes. Understanding how to define these spaces and translate and transform them in time is necessary and requires concepts from fundamental geometry.

Closest Point of Approach: There are various forms of the Closest Point of Approach problem [56]. The proposed ambiguity resolution algorithm requires solving of the simplest version of this problem in order to check for potential conjunctions between signals: analyzing two points moving at constant velocities.

Lattice Algorithms and Geometry of Numbers: The intersection of multiple wavefronts creates lattices of points that serve as candidate solutions for observer locations. Understanding the geometric configurations between these points and being able to measure the distance between them leverages lattice algorithms [57] as well general concepts from the geometry of numbers [58], [59].

Timing Noise and Uncertainty: Timing uncertainty associated with the signal processing of pulses has a large impact on the number of potential solutions and thus plays a key role in establishing areas of potential intersection between signals and probabilities of candidate solutions being present in those areas.

CHAPTER VI

X-RAY PULSAR-BASED NAVIGATION - RESOLVING AMBIGUITY AND ESTIMATING POSITION

The algorithm proposed in this chapter seeks to discover the minimal time interval $T_{\min}$ and distance $D_{\min}$ between candidate solutions and then provide a position estimate associated with each solution. It works by first only checking the intersections between pulsar pairs that have the lowest periodicity of intersection. Then as candidate solutions sets are investigated, those that satisfy conjunction criteria are saved. Over multiple iterations the algorithms seeks to filter out candidate solution sets which don't have a sufficiently tight intersection of pulse waves. After filtering and discarding, the remaining solutions are used to perform position estimation. If more than one solution remains, the minimal time interval and distance between the remaining solution pairs is considered to be the minimal interval for the given search parameters.

Successful verification of the algorithm would entail that for a given known drift in time and uncertainty in position the algorithm will successfully return all candidate solutions and provide a true estimate of the distance between the minimal solution pairs. Failure of the algorithm would be to omit intersections and produce no candidate solution when the space being considered is large enough in time and space to contain the true observer location. A partial successes would be to discover candidate solutions, but not be able to ensure that all possible solutions have been discovered and/or to be unable to guarantee that the estimated $T_{\min}$ and $D_{\min}$ are global minima. If a lower bound for $T_{\min}$ and $D_{\min}$ is not set, then the trustworthiness of any proposed solution estimates is put into question.

A full detailing of the algorithm architecture follows. Potential shortcomings and limitations of the method are also addressed. Empirical testing is needed in order to verify functionality of the algorithm and whether or not proposed solution pairs constitute global minima pairs. It is unknown currently if the minimal time and space solutions can be found using deterministic means, however, initial investigation towards a deterministic inductive method is also proposed. Testing procedures are proposed for estimating the probability of false positives (perceiving and failing to filter out a candidate solution when there isn't one), and false negatives (omission of candidate solution).

Algorithm Architecture

To understand the meaning behind the variables and terms used in the algorithm, reference the nomenclature provided before the introduction of this dissertation. Referencing Figure VI.1 one can see that besides a few filtering loops, the algorithm follows a linear progression along a series of steps.

The assumptions that go into the architecture of this algorithm are presented next followed by a description of each of the steps shown in Figure VI.1.

Assumptions of Algorithm Construction

The following assumptions are made in the construction of the algorithm. Some of them can be relaxed in the future in order to increase the capabilities of the

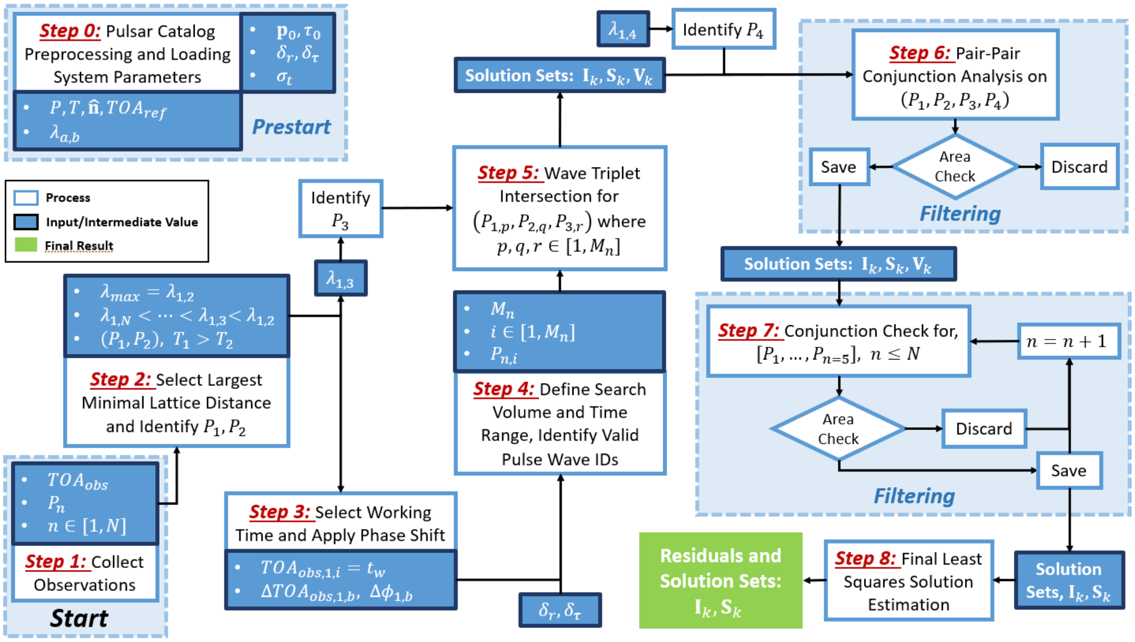

Step 0: Pulsar Catalog Preprocessing and Loading System Parameters
$\mathbf{p}_0, \tau_0$
δ_r, δ_τ
σ_t
$P, T, \hat{\mathbf{n}}, TOA_{ref}$
$\lambda_{a,b}$
Prestart
Process
Input/Intermediate Value
Final Result
$\lambda_{max} = \lambda_{1,2}$
$\lambda_{1,N} < \cdots < \lambda_{1,3} < \lambda_{1,2}$
$(P_1, P_2),\ T_1 > T_2$
Step 2: Select Largest Minimal Lattice Distance and Identify P_1, P_2
TOA_{obs}
P_n
$n \in [1, N]$
Step 1: Collect Observations
Start
Identify P_3
$\lambda_{1,3}$
Step 3: Select Working Time and Apply Phase Shift
$TOA_{obs,1,i} = t_w$
$\Delta TOA_{obs,1,b},\ \Delta\phi_{1,b}$
$\lambda_{1,4}$
Identify P_4
Solution Sets: $\mathbf{I}_k, \mathbf{S}_k, \mathbf{V}_k$
Step 5: Wave Triplet Intersection for $(P_{1,p}, P_{2,q}, P_{3,r})$ where $p, q, r \in [1, M_n]$
M_n
$i \in [1, M_n]$
$P_{n,i}$
Step 4: Define Search Volume and Time Range, Identify Valid Pulse Wave IDs
δ_r, δ_τ
Step 6: Pair-Pair Conjunction Analysis on (P_1, P_2, P_3, P_4)
Save
Area Check
Discard
Filtering
Solution Sets: $\mathbf{I}_k, \mathbf{S}_k, \mathbf{V}_k$
Step 7: Conjunction Check for, $[P_1, \ldots, P_{n=5}],\ n \leq N$
$n = n + 1$
Area Check
Discard
Save
Filtering
Residuals and Solution Sets: $\mathbf{I}_k, \mathbf{S}_k$
Step 8: Final Least Squares Solution Estimation
Solution Sets, $\mathbf{I}_k, \mathbf{S}_k$

Figure VI.1: Overall flowchart showing how ambiguity can be quantified and resolved, and position and time be estimated.

algorithm in exchange for computational complexity and robustness.

2D Planar Wave Problem: Pulsars ar considered to be planar, only two spatial dimensions are considered, so wavefronts are represented by infinite lines that move through the field at c. Note that these lines do *not* represent planar wavefront represented in a 3D space intersecting with a working plane. That would allow for lines to move at speeds less than c in two dimensions, depending on the direction of propagation with respect to the working plane. This is not the case, and thus all lines move with velocity $c\hat{\mathbf{n}}$ in two dimensions. It is expected that the algorithm can be extended to n dimensional space, and thus restricting the analysis to two dimensions allows for a simpler architecture for proof of concept without loss of generality.

Stationary Observer: This is a simplifying assumption which reduces the complexity of the problem. It is possible to still resolve the ambiguity problem and determine a position estimate for a non stationary observer, however this adds complexity which will not be studied here.

Non-Parallel Wavefronts: It is assumed that no two pulsars have signals which come from *exactly* the same direction, and thus unique intersections can be calculated for the lines that represent the wavefronts from each pulsar.

Observed Pulsars Are Readily Identifiable: All relevant information associated with a pulsar is known. Pulsar ID, period, phase, direction of propagation, etc. is known immediately once a signal is resolved.

Simultaneous Observation of Multiple Pulsars is Possible: There exists a time window which contains multiple TOA_{obs}s from all observed pulsars. TOAs from various pulsars can either be observed during this time window, or if a TOA lies outside this time window, it can be mapped into the time window using accurate knowledge of pulsars periods if the clock drift is assumed to be the same for observations during the time window and the TOA which lies outside of it.

Pulse Signals Provide TOAs with Fixed Time Uncertainty and Equal Clock Drift: Observations provide a TOA_{obs} associated with each pulsar. All $TOAs$ are affected by the same constant clock drift and all experience timing noise sampled from a normal distribution of fixed standard deviation σ_t. This timing noise model is simplified later by assuming that the probability of timing error is actually uniform over the interval $[0, \sigma_t]$.

TOA Alignment at True Solution: The true solution needs to exist at a single point in space-time which will be found by aligning the observation phases. However, in setting this alignment up, we may be neglecting timing noise, and thus potentially neglecting candidate solutions.

Step 0 – Pulsar Catalog Preprocessing and Loading System Parameters

Before the process of X-Nav and ambiguity resolution can begin a pulsar catalog must be loaded. This catalog, or subcatalog, contains relevant information regarding all pulsars P which would potentially be of interest to an observing spacecraft. The periods T, directions of propagation $\hat{\mathbf{n}}$, and phases ϕ necessary for calculating TOA_{ref} are provided. The minimal lattice distances $\lambda_{a,b}$ between pulsar pairs are also precomputed and provided.

System parameters pertaining to the observer are also necessary. An initial estimate for the observer position $\mathbf{p}_0$ and initial estimate of observer time τ_0 is loaded. Estimates of position and clock uncertainty, δ_r and δ_τ are needed as well. Finally the timing uncertainty σ_t associated with how accurately the timing of a signal can be resolved by signal processing also must be provided.

Step 1 – Collect Observations

Once an observer has loaded their catalog and has estimates of their system parameters they can collect observations. When pulsars are observed and identified signal processing will resolve the incoming light and give a local time-of-arrival for each observed pulse of the observed pulsars $TOA_{obs,n,i}$. Each observed pulsar is labeled as P_n where $n \in [1, N]$ and N is the total number of observed pulsars. For each pulsar P_n there are M_n observed pulse wavefronts, and each one is given an ID $P_{n,i}$ where $i \in [1, M_n]$.

Step 2 – Find Largest Minimal Lattice Distance and Identify P_1 *and* P_2

For the observed pulsars P_n search the catalog for minimal lattice distance $\lambda_{a,b}$ and identify the largest minimal lattice distance $\lambda_{\max} = \lambda_{1,2}$. The existence of these minimal distances is proven by Minkowski's Theorem and these distances can be done through lattice algorithms. This serves to identify P_1 and P_2, which will be the two pulsars which have the largest minimal lattice distance among all possible observed pulsar pairs. Of those two, P_1 has the longer period: $T_1 > T_2$. Collect all of the other minimal lattice distances for the remaining pulsars paired with P_1 so that $\lambda_{1,2} > \lambda_{1,3} > ... > \lambda_{1,N}$. From this ordering identify P_3 associated with $\lambda_{1,3}$.

Step 3 – Select Working Time and Apply Phase Shift

The next step is to identify the working time: $t_w = TOA_{obs,1,i}$. This time will be associated a peak arrival time for one of the pulse waves coming from pulsar P_1. Which pulse number is chosen will depend on the time of interest t^*. Since observation, signal processing, and resolving of pulsar signals is a process which takes hundreds or thousands of periods from the incoming pulse, it is up to the observer to decide at exactly which time during the observation window would they like to estimate position. Whatever time of interest is chosen, the nearest associated TOA from pulsar P_1 is defined as $TOA_{obs,1,i} = t_w$ so that $|t^* - t_w| < T_1$. Once this working time is selected, a phase shift $\Delta\phi_{1,b}$ is applied to all observed $TOA_{obs,b,i}$ from other pulsars so that one wavefront from every other pulsar lines up with $TOA_{obs,1,i}$. This phase shift is demonstrated in Figure VI.2. The goal of this shift is to help the observer find a point in time (and space) where all wavefronts intersect since this represents a candidate solution for a position estimate.

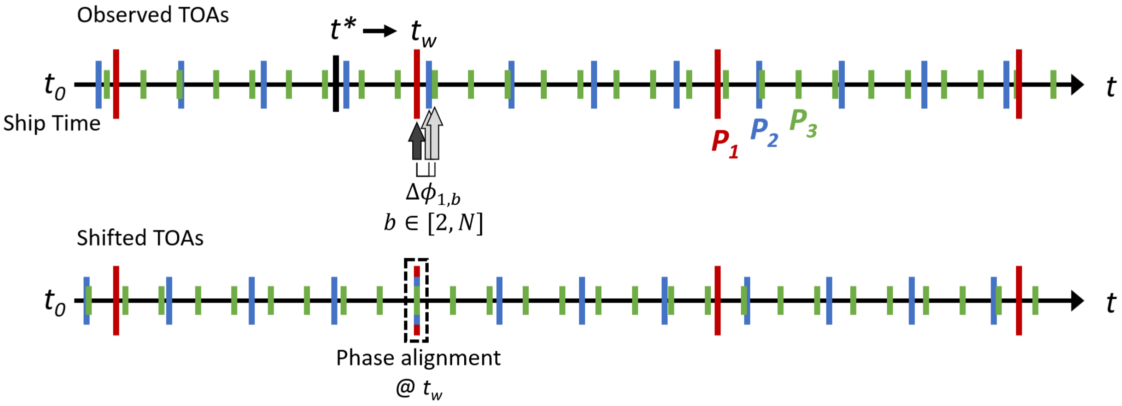


Figure VI.2: Selection of t_w and phase shift $\Delta\phi_{1,b}$ applied to all TOAs from pulsars P_2 through P_N.

These phase shifts are also applied in the spatial domain. All wavefronts for pulsars P_b for $b \in [2, N]$ are shifted by a distance $c\Delta t_b \hat{\mathbf{n}}_b$, where Δt_b is the time shift associated with $\Delta\phi_{1,b}$ and c is the speed of light. This is shown in Figure VI.3. This shift will hopefully produce intersections between pulsars signals where one pulse from every single observed pulsar meet to create a possible candidate solution for a position estimate. It's possible that before the shift there were locations where an intersection pulse wavefront from all observed pulsars existed. These locations would represent places in time and space where bright peaks from all pulsars could be seen simultaneously. However, they would only represent candidate solutions if the observer actually observed simultaneous peaks from all pulsars. It's more likely that no such solution candidates will exist, until the phase shift is applied, since the phase shift moves the wavefronts so that it's possible for there to be a point in space and time where signals come together in a way that is similar to what the observing spacecraft witnessed in its observations.

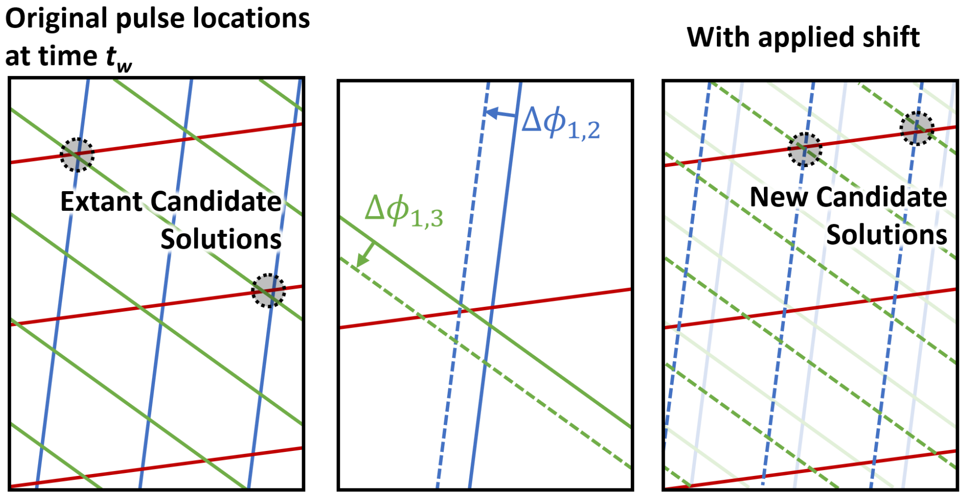


Figure VI.3: Spatial shifting of wavefronts by $c\Delta t_b\hat{\mathbf{n}}_b$. Note that some extant candidate solutions may be eliminated by the shift and new ones may be created.

Step 4 – Define Search Area and Time Range, Identify Valid Pulse Wave IDs

Once phase shifts have been applied to all pulses, the next step is to define the search range for the position estimation problem. The system parameters δ_τ and δ_r which represent estimated uncertainty of the observer's clock and position are needed. A reasonable minimal time range for searching for potential solutions would be $t_w \pm \delta_\tau/2$ and a reasonable minimal search space would be the area of the circle (or sphere, in three dimensions) with radius $\delta_r/2$ centered on position estimate $\mathbf{p}_0$. The larger the search space, the more computationally complex the problem of ambiguity resolution becomes. In the case where no candidate solutions are found in the designated search space, and iterative approach is possible. Where the search space is increased until one candidate solution, or some minimal number of candidate solutions, are generated. Such an approach would be necessary in cases where clock drift and/or position uncertainty are far worse than expected.

Once the search space is defined, the value M_n is calculated for each observed pulsar P_n. M_n is the total number of unique pulses that enter the spatial range over the designated time range. Pulsars with higher frequencies will have higher M_n values, thus it is expected that $M_N \geq ... \geq M_n \geq ... \geq M_1$. These M_n values can be easily calculated by knowing the direction and speed of propagation, c, as well as the periods of each pulsar signal. Once these are calculated, a counting index $i \in [1, M_n]$ is associated to every single wavefront in the search space. These wavefronts are identified by the pulse wave ID $P_{n,i}$ where n indicates which pulsar the signal belongs to, and i indicates the number of that wavefront.

Step 5 – Wave Triplet Conjunction Analysis

The intersection of two wavefronts from two different pulsars can be modeled as the intersection between two non-parallel lines. It is guaranteed that at every point in time there will be an intersection between these lines. If a third pulsar is considered, then a third line which is not parallel to either of the first two is added.

These lines do not intersect at all moments in time, however as long as they are all moving relative to one another, it is guaranteed that there exists a point in time where all three will intersect. Finding these intersections is the purpose of the wave triplet conjunction analysis.

Since the computational complexity of the ambiguity problem is reduced when there are fewer candidate solutions which are represented by intersections, triplet conjunction analysis is performed on the signals from the three pulsars which have the largest minimal lattice distances. P_1 and P_2 have already been defined. P_3 is the pulsar which has the second largest minimal lattice distance $\lambda_{1,3}$ when paired with P_1 Thus, the pulse waves considered in the triplet conjunction analysis are $(P_{1,p}, P_{2,q}, P_{3,r})$ where p, q, r are the indices $p \in [1, M_1]$, $q \in [1, M_2]$, and $r \in [1, M_3]$.

Every single possible triplet of pulse waves creates a valid solution candidate with a location in space and time of conjunction: (x_c, y_c, t_c), thus the total number of candidate solutions produced at this step is $N_{sol} = M_1 \times M_2 \times M_3$. Every solution $\mathbf{S}_k = (x, y, t)_k$ has an ID associated with it $\mathbf{I}_k$, where $k \in [1, N_{sol}]$ where N_{sol} is the number of current candidate solutions. The ID $\mathbf{I}_k$ is composed of the set of wave number IDs which simultaneously overlap for that solution. N_{sol}, $\mathbf{I}_k$, and $\mathbf{S}_k$ will change value as solution candidates are filtered out and updated. $\mathbf{I}_k$ will grow as more waves pulse ID's are associated with the same solution candidate, starting off initially as just $\mathbf{I}_k = (P_{1,p}, P_{2,q}, P_{3,r})$ and eventually at the last step containing a wave from every observed pulsar $\mathbf{I}_k = (P_{1,p}, P_{2,q}, ..., P_{N-1,r}, P_{N,u})$. $\mathbf{S}_k$ currently represents an exact mathematical answer to the question of where three lines intersect in time and space. In later steps, as more pulse waves are associated with a solution candidate this exact answer will be replaced by midpoint or least square solutions.

The time and location of triplet conjunction is calculated by first selecting the index, p, q, r of the wavefronts. Then, using the working time t_w as a reference, the location of the intersection of $P_{1,p}, P_{2,q}$, is found. This is done by representing each front by a line and finding their intersection. This is a virtual point with velocity $\mathbf{v}$. To find the distance from this point to $P_{3,r}$, a line must be defined which represents the wavefront, after which it's a simply matter of calculating the distance from a point to a line. The relative velocity between this point and the line of $P_{3,r}$ determines when and where the triplet conjunction occurs. The dot product between $\mathbf{v}$ and $\hat{\mathbf{n}}_3$ provides the component of $\mathbf{v}$ which is perpendicular to the wavefront (which is also moving) and allows for the calculation of the time elapsed (positive or negative) between triplet conjunction and working time t_w.

Duration and Bounds of Conjunction: Once the time and location of triplet conjunction is found, (x_c, y_c, t_c), it is also necessary to determine for how long this conjunction is valid. Due to timing noise σ_t each wavefront is more accurately represented by a band of finite thickness, instead of a line. Thus, while the exact conjunction might occur at an infinitesimal moment in time and space, the three bands come together to create an overlapping area which endures for a finite period of time as the bands come together to a perfect conjunction and then eventually separate, going through the various stages of overlap that are shown in Figure V.7. The center of the overlapping area also moves through space, potentially over large distances between the instance where it first appears and the last moment when it becomes an infinitesimal point of overlap and then vanishes. This fact is relevant when searching for other pulsar signals which may also intersect with this overlap area.

In order to determine for how long the conjunction lasts, the three intersection points and their corresponding areas are tracked. The locations of the three intersection points A, B, C, are found by solving for the intersection between the lines of each pulse wave. These points define a triangle which shrinks as the lines converge on its incenter before conjunction, and then grows as the wave fronts continue on passed the conjunction. The lines will always meet at the incenter of the triangle, however depending on the geometric configuration of the pulse waves, the location of the incenter $(x_c(t), y_c(t))$ may move over time. To calculate the location of the incenter Equation 6.1 is used

$$x_c(t) = \frac{aA_x + bB_x + cC_x}{a+b+c} \qquad y_c(t) = \frac{aA_y + bB_y + cC_y}{a+b+c} \tag{6.1}$$

where A_x, A_y are the time varying coordinates of intersection point A, etc. and a, b, c are the time varying side lengths opposite the vertexes A, B, C.

To understand how long the conjunction lasts, and what space it covers, it is necessary to know the velocity, $\mathbf{v}_c$, of the incenter of the triangle, as well as know the velocities of each of the corner points relative to the incenter point. These velocities can be easily calculated by taking the derivatives of the time dependent locations of the intersection points and the incenter point. Since the wave fronts move at constant velocities all derived velocities will also be constant.

These relative velocities, $\Delta\mathbf{v}_i = \mathbf{v}_i - \mathbf{v}_c$, are necessary in order to determine how quickly intersection point moves away from the conjunction. Using the relative velocities, it is able to determine how much time elapses from conjunction before the distance from each intersection point to the conjunction location is too great for the possibility of overlapping areas from all three bands. The process is illustrated in Figure VI.4.

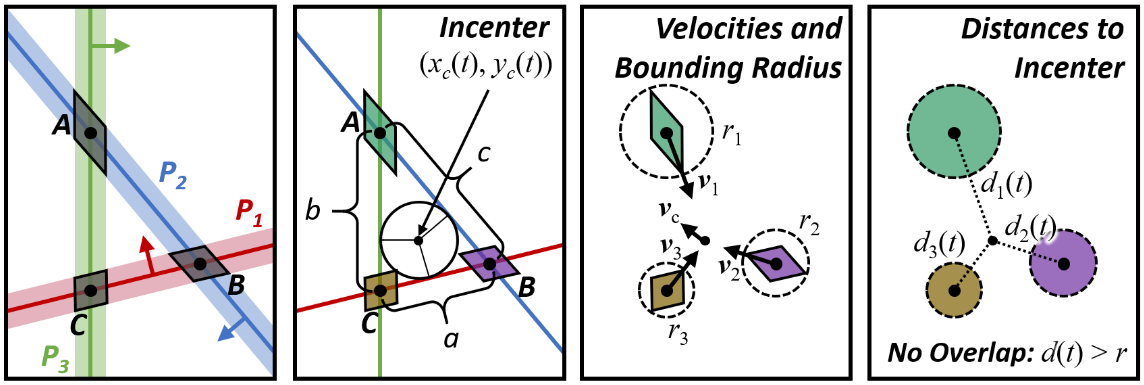


Figure VI.4: Steps for finding distance from intersection points to conjunction point for three pulse waves. **Left to Right:** Three points are defined by intersecting waves. The points define a triangle with an incenter at (x_c, y_c). Each point has a velocity $\mathbf{v}$ and associated a bounding radius r for the area created by the overlapping bands of the wave fronts. The distance from the points to the incenter is $d(t)$.

Ever intersection point is associated with an area of intersection. These areas can be bound by a radius r by calculated and halving the long diagonal of a parallelogram. Since all three intersection points converge perfectly at the conjunction point, the amount of time it takes for each of these points to traverse a distance r from the conjunction point is the amount of time during which there will be solution candidate where bands from all three pulse wavefronts overlap. After that (or before, if analyzing the problem before conjunction) no overlap is possible. Thus the maximal duration of the conjunction is calculated according to Equation 6.2

$$t_{\max} = \min\left(\frac{||\Delta v_i||}{r_i}\right), \quad i \in [1,3] \tag{6.2}$$

The subscript "max" indicates that it is the maximal amount of time which can elapse before one of the three intersection points, or bounding triangle vertexes, is too far away for there to be an overlap between all three wave front bands.

The last piece of necessary information for characterizing the nature of the triplet overlapping areas is the size of area. As a conservative estimate, which removes that chance of omitting potential conjunctions, the area can be slightly overestimated by assuming it to be a circle of radius $r_{\max} = \max(r_i)$ where r_i is the radius associated with the areas around each of the three intersection points. The true overlapping area created when all three points converge will be smaller than this.

Now, the time of conjunction, maximal overlapping area size, and duration of solution have been estimated. Using this information a large bounding area can be defined which encompasses all possible points where there might be a potential triplet conjunction over the duration of the conjunction. This bound of this area will be used in the next step in order to select candidate waves from another pulsar which may also form a conjunction with triplet from this step. The complete bounding area associated with the conjunction is depicted in Figure VI.5

The four vertexes of the bounding conjunction area parallelogram are labeled as V_1, V_2, V_3, V_4. Vertexes V_1 and V_2 can be calculated by finding the locations $x_c(t_c - t_{\max}), y_c(t_c - t_{\max})$ and $x_c(t_c + t_{\max}), y_c(t_c + t_{\max})$. Vertexes V_3 and V_4 can be calculated by using the radius $r_{\max}$ of the inscribed circular area and knowledge of the direction

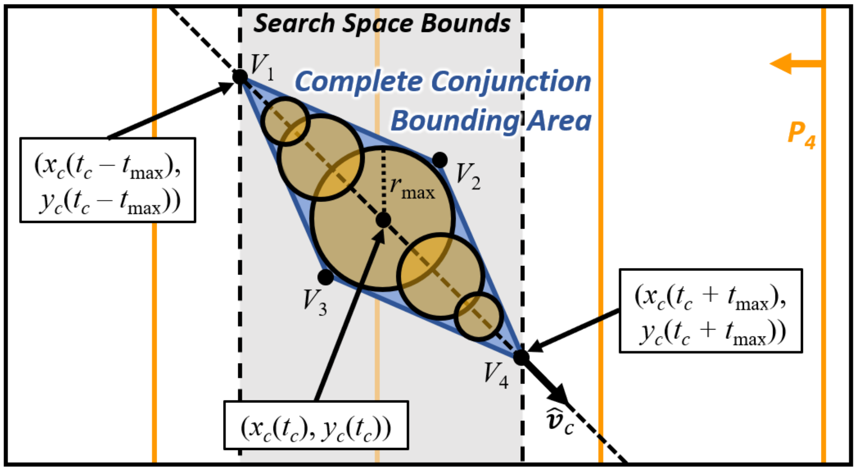

Search Space Bounds
Complete Conjunction Bounding Area
P_4
V_1
V_2
V_3
V_4
$r_{\max}$
$(x_c(t_c - t_{\max}), y_c(t_c - t_{\max}))$
$(x_c(t_c + t_{\max}), y_c(t_c + t_{\max}))$
$(x_c(t_c), y_c(t_c))$
$\hat{v}_c$

Figure VI.5: Complete area which bounds the space of potential overlap between three pulse waves over the duration of a conjunction. The area is defined by a parallelogram with vertexes V_1, V_2, V_3, V_4

of $\hat{\mathbf{v}}_c$. These four spatial coordinates must be recorded along with the candidate solution. Thus $\mathbf{S}_k$ now also is accompanied by $\mathbf{V}_k = (V_1, V_2, V_3, V_4)_k$ as well as the first moment when the bounding area appears $t_0 = t_c - t_{\max}$ and last moment it disappears $t_1 = t_c + t_{\max}$. This information will be necessary for checking if pulse waves from other pulsars might potentially form a conjunction with the existing triplet conjunction. The final outputs of this step will be $\mathbf{I}_k = (P_{1,p}, P_{2,q}, P_{3,r})_k$, $\mathbf{S}_k = (x_c, y_c, t_c, t_0, t_1)_k$, and $\mathbf{V}_k = (V_1, V_2, V_3, V_4)_k$.

Step 6 – Pair-Pair Conjunction Analysis

First, P_4 is identified by selecting the pulsar which forms the minimal lattice distance $\lambda_{1,4}$ when paired with P_1. Pair-pair conjunction analysis is then performed using pulse waves from the pulsars (P_1, P_2, P_3, P_4). This process is started by using the candidate solutions $\mathbf{I}_k$ and their bounding areas provided by *Step 5* and solutions which satisfy the conjunction analysis are kept while others are discarded. Also, a new bounding area for each new solution $\mathbf{I}_k = (P_{1,p}, P_{2,q}, P_{3,r}, P_{4,u})_k$ is calculated at the end of the pair-pair analysis which represents the conjunction area of the four pulse waves.

At the beginning of *Step 6*, a given solution ID $\mathbf{I}_k$ is composed of three pulse wave ID's: $(P_{1,p}, P_{2,q}, P_{3,r})$ and has an associated bounding area defined by the vertices $\mathbf{V}_k$. This area first appears at time t_0 and lasts until time t_1. All pulse waves of P_4 which might potentially intersect this area during the interval of time (t_0, t_1) must be selected as candidates for conjunction analysis. Even though the true conjunction area of $(P_{1,p}, P_{2,q}, P_{3,r})$ does not fill the entire bounding area for the interval (t_0, t_1), treating the entire area as a viable area of conjunction for the full interval simplifies analysis, at a marginal increase of computational complexity, in the form of additional checks which must be performed.

The first waves which can immediately be selected as candidates for analysis are the ones which are already intersecting the area at time t_0. In addition to these waves, all P_4 waves which are not intersecting the area at t_0, but are at a distance $d < c \times (t1 - t0)$ from the closest vertex V must also be selected for conjunction analysis, since they will eventually intersect the area before it disappears.

Once all the candidates have been selected, pair-pair conjunction analysis can be done sequentially on each. The ordering doesn't matter, since each conjunction with different pulse waves is independent of others. Suppose the earliest pulse wave from P_4 to enter the bounding area is used to start the checking sequence. This pulse wave has an ID $P_{4,u}$, where u takes on some value $u \in [1, M_4]$, however since only one wave from any pulsar is being considered at a time, the second subscripts will be temporarily omitted from all pulse wave IDs to simplify expressions. Once the conjunction check with $P_{4,u}$ is complete, then next wave $P_{4,u+1}$ is selected and the check is performed again until all candidates have been exhausted for the current set of waves provided from *Step 5* $\mathbf{I}_k$. After which, the next candidate solution from *Step 5*, $\mathbf{I}_{k+1}$, is selected and another series of pair-pair conjunction checks is performed.

Thus, for a selected solution $\mathbf{I}_k$ from *Step 5*, and a selected candidate from P_4 the four lines associated with each of the four pulse waves can be defined using simplified notation: P_1, P_2, P_3, P_4. These four lines are shown in the first panel of Figure VI.6 and can be intersected in pairs to create three unique sets of point pairs, as listed in Equation 6.3.

$$Pairings(P_1, P_2, P_3, P_4) = \left\{ \begin{matrix} (P_1, P_2), (P_3, P_4) \\ (P_1, P_3), (P_2, P_4) \\ (P_1, P_4), (P_2, P_3) \end{matrix} \right\} \tag{6.3}$$

Considering the first possible point paring: $(P_1, P_2), (P_3, P_4)$ this defines two points formed by the intersections of two pairs of lines. Each line is moving with constant speed c with some direction $\hat{\mathbf{n}}$. Thus, each of the two intersection points is also moving at a constant velocity, $\mathbf{v}_1$ and $\mathbf{v}_2$. In order to determine if the two points meet at a conjunction, the *Closest Point of Approach* (CPA) problem must be solved. It can be easily solved analyticity using non-iterative methods based on linear algebra [56] and the solution returns the time of closest approach t_{ca} between the two intersection points as well as the minimal distance between the two points $d_{\min} = d(t_{CA})$. In order to determine if a conjunction has occurred, the timing noise σ_t is used to establish the width of the band associated with the pulse waves. These bands overlap to define a bounding area around each intersection point which represents the space of possible locations for where the intersection between the two waves could occur. Figure VI.6 illustrates how the areas are defined and the final conjunction check is performed.

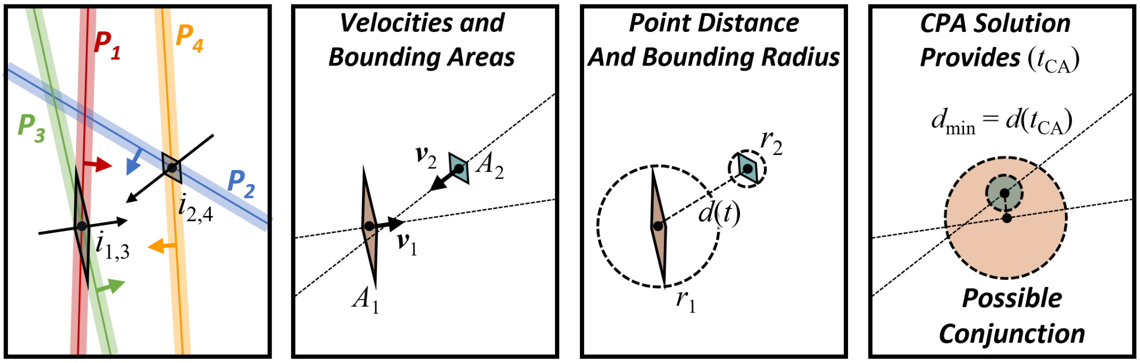


Figure VI.6: Illustration of how conjunction is determined when analyzing a two points defined by pulse wave pairs. **Left to Right:** Initial points defined by intersecting waves. Each point has a velocity $\mathbf{v}$ and associated area A created by overlapping bands of the wave fronts. The distance between points is $d(t)$ and the areas are bound be circles of radius r. At time of closest approach t_{CA} the bounding circles overlap, thus a possible conjunction is flagged.

If the points happen to come together close enough so that their corresponding areas A might intersect, then a possible conjunction is flagged. Each area A is bound by a circumscribing circle of radius r. Conjunction checks are performed using these bounding circles, instead of the areas A. The reason for this substitution is that it's possible for there to be overlap between areas A_1 and A_2 at times other than t_{CA} while having no overlap at time t_{CA}. In order to insure that no possible conjunction is missed (a false negative result) a more complex conjunction check between areas A_1 and A_2 would be necessary, one which possibly involves using a numerical approach with small time steps checking for intersecting areas over a time range. By using circles, not only is it computationally more simple to check for area intersections, but there is also no chance of missing a possible conjunction when checking for potential intersection at time t_{CA}. However, since the intersection of circles does not guarantee intersection of areas A_1 and A_2, conjunction flags indicate the *possibility* of a conjunction only. If at time t_{CA} Inequality 6.4 is satisfied,

$$d_{\min} > r_1 + r_2 \tag{6.4}$$

then there is no chance of there ever being a potential conjunction between the two points; otherwise, a conjunction is possible. The drawback of this approach is the inclusion of false positive candidate solutions, which adds computational complexity when performing additional filtering of candidate solutions in later stages of the algorithm.

If a potential conjunction has been detected, then, similar to *Step 5*, the duration and spatial bounds of the potential conjunction must be determined. A coarse estimate of the entire bounding bounding area for the pair-pair conjunction can be estimated by taking the area of the parallelogram which is formed when intersecting the swaths of area created by the circles associated with each intersection point, as shown in Figure VI.7. The method for identifying the locations of the vertexes of this parallelogram is identical to the method for identifying the vertexes of the parallelograms creates when intersecting the bands of two wavefronts.

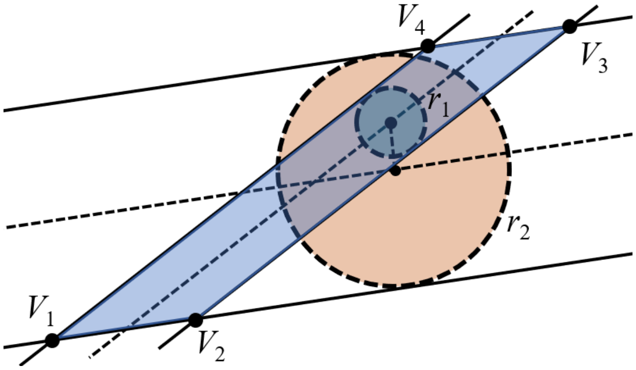


Figure VI.7: Bounding parallelogram with vertexes V_1, V_2, V_3, V_4 defined by overlapping paths of bounding circles of two intersection points. Area bounds the possible locations for a conjunction by the four pulse waves being analyzed.

This bounding area does not take into account the timing of the closest approach, and is thus not a tightly fitting bound. A more accurate complete conjunction bound may be devised if the locations where intersection of the two bounding circles associated with the two intersection points is no longer possible, i.e. the locations where the overlapping circles separate. At these locations the distance between the two intersection points is $d_{max} = r_1 + r_2$, as shown in Figure VI.8.

At this moment, the two circles touch tangentially, and that point of contact is the location of the last possible intersection between the bands of the four pulse waves being studied. It is necessary to find this location to bound the area of possible conjunction. To find this point, first consider Figure VI.9. This figure depicts the change in distance between intersection points. That distance is minimal at time

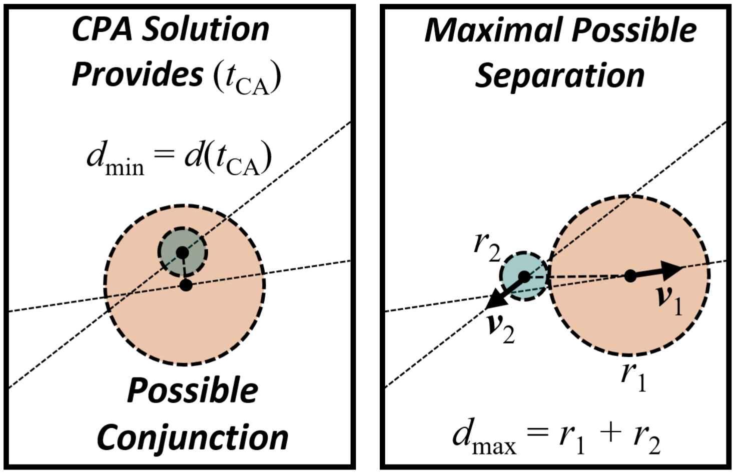

CPA Solution Provides (t_{CA})
$d_{\min} = d(t_{\mathrm{CA}})$
Possible Conjunction
Maximal Possible Separation
r_2
v_1
v_2
r_1
$d_{\max} = r_1 + r_2$

Figure VI.8: $d_{\max}$ is the maximal distance between intersection points which allows there to be overlap between the areas described by r_1 and r_2.

t_{ca} and increases as time moves forward or backward from that moment. At some point, the distance between the two intersection points, which can be measured as the distance between a point on the line moving with velocity $\mathbf{v}_c$ and the origin, will reach $d_{\max}$.

A right triangle is formed by d_min and d_max which allows the calculation of the distance along the line, which will be called d_l. It is necessary to calculate the amount of time which a point moving along the line, which represents the varying distance between the two intersection points, will traverse d_l. This is given by Equation 6.5.

$$\Delta t = \frac{d_l}{||\mathbf{v}_c||} = \frac{\sqrt{d_{\max}^2 - d_{\min}^2}}{||\mathbf{v}_c||} \tag{6.5}$$

Thus, at time $t_0 = t_{ca} - \Delta t$ and $t_1 = t_{ca} + \Delta t$ the two circles are tangent. Using knowledge of the time at which this occurs the location of each intersection point can be determined. Then, the line connecting the two points can be defined, and starting

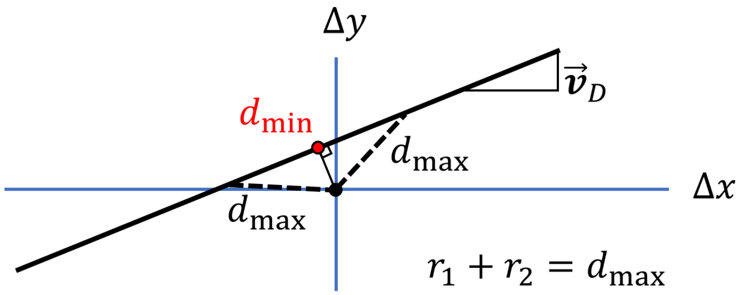


Figure VI.9: Parametric line representation of the distance between the two intersection points. The minimal distance between the line and origin $d_{\min}$ is the minimal distance between the intersection points at the time of closest approach, t_{ca}. At times $t_{ca\pm\Delta t}$ the distance between the points is $d_{\max}$.

at one of the points, move along the line by the radius of the bounding circle r. That will be the location of the tangent point between the two circles at that moment in time. The two tangent points associated with time $t_0 = t_{ca} - \Delta t$ and $t_1 = t_{ca} + \Delta t$ can serve as the refined vertexes V_1 and V_4 of the complete conjunction bounding area. A form similar to that shown in *Step 5* is used here. The parallelogram has a circle inscribed in it which represents the maximal size of the overlapping area created by the two intersection points during closest approach. One can take the time to compute this area, or simply assume it to be the size of the smaller of the two circles. The center of this overlap area at time t_{ca} is the center of the bounding parallelogram. Having the locations of V_1 and V_4, as well as the radius and center of the inscribed circle, one can defined the other two vertexes V_2 and V_3. If it turns out that the distance between V_1 and V_4 is less than $2r_{\min}$ then, the complete conjunction bounding area can be bound by just a circle of radius $r_{\min}$ centered on the centroid of the intersecting area of the two circles at closest approach.

The procedure listed above is performed for each of the three possible point pairs defined by P_1, P_2, P_3 and $P_4 = P_{4,u}$ and then again three more times for the point pairs created by the remaining candidate pulse waves of P_4. At the end of *Step 6* the remaining candidate solutions have IDs in the form $\mathbf{I}_k = (P_{1,p}, P_{2,q}, P_{3,r}, P_{4,u})_k$ and solutions $\mathbf{S}_k = (x_c, y_c, t_{CA}, t_0, t_1)$ where (x_c, y_c) represent the coordinates of the centroid of overlapping area between the two bounding circles at time t_{ca}, and $\mathbf{V}_k = (V_1, V_2, V_3, V_4)_k$.

Step 7 – Conjunction Check Including P_5 *...* P_N

Step 5 and *Step 6* of the ambiguity resolving process attempt to narrow down the spatial and time range where candidate solutions could occur. They also aim to reduce the combinatorial complexity which would result from trying to check all wave fronts from all pulsars for potential intersection without filtering. *Step 7* does not narrow down the search space any further. For each candidate solution $\mathbf{I}_k$ and $\mathbf{S}_k$ passed on from *Step 6* this step iterates through all the remaining pulse waves from P_5 through P_N and checks which ones might potentially intersect the complete conjunction bounding areas which were calculated in the previous steps. Pulse waves with bands which intersect them are kept as members of potential candidate solutions, while the remaining ones are discarded. Thus, at the end of this process the relevant solution ID sets will be $\mathbf{I}_k = (P_{1,p}, P_{2,q}, ..., P_{N,r})_k$. Hopefully, at this stage of the filtering process there are very few candidate solutions; ideally, there would only be one left. If the timing noise is large, then there will potentially be many candidate solutions identified. If no candidate solutions are found, then the the time and space domain in which the search is occurring might need to be expanded.

Step 8 – Final Solution Estimation

The final solutions estimate will be performed using least squares, since it is highly improbable that all pulse waves will perfectly intersect each other at a moment in time. Least squares is performed here using an implicit formulation of a line to represent a pulse wave, which can be constructed at any point in time for any pulse wave using the direction of propagation $\hat{\mathbf{n}}_n$, period T_n and the pulse wave ID $P_{n,i}$, however other representations are also valid. Each wave front is moving with a constant velocity which is accounted for with a linear time term. Thus, the implicit form of the line representing a moving wavefront is shown in Equation 6.6.

$$D_n = A_n x + B_n y + C_n t, \quad n \in [1, N] \tag{6.6}$$

To get values for A, B, C, D for each line, $t_w = 0$ can be used as a reference time. Thus, t would represent time since, or before t_w. We write the system of equations representing all lines in a solution set $\mathbf{I}_k$ in the form $\mathbf{d}_k = \mathrm{A}_k \mathbf{x}_k$:

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_N \end{bmatrix}_k = \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ \vdots & \vdots & \vdots \\ A_N & B_N & C_N \end{bmatrix}_k \begin{bmatrix} x \\ y \\ t \end{bmatrix}_k$$

and can apply the pseudo-inverse, $(\mathrm{A}^{\mathrm{T}}\mathrm{A})^{-1}\mathrm{A}^{\mathrm{T}}\mathbf{d} = \mathbf{x}$, to solve for the least squares solution (x, y, t), which would give the point in space and time which is closest to all pulse waves belonging to $\mathbf{I}_k$. Here, solutions which are too distant from any pulse waves in the set $\mathbf{I}_k$ can be discarded, if at time t the distance between the least squares solution x, y and any of the lines is greater than $c\sigma_t$, then the bands associated with each line don't all overlap, and thus there can't be a solution. Alternatively, all least squares solutions can be kept and fed into a navigation filter which weighs the residuals associated with each solution, and selects the solution with the lowest residuals.

Under nominal operation, there should always be a solution left after filtering, since the observer must exist *somewhere* in time and space. In the case of multiple solutions remaining at the end of the process, a navigation filter can help eliminate false candidate solutions. Alternatively, system parameters may need to be updated or re-estimated. If no solutions are remaining, or the ones produced by the algorithms are not satisfactory, such as having too high a residual, then it may be necessary to expand the time and space search domain.

Alternative Proposed Methods

A method of using spherical volumes to partition the search space and speed up the filtering of candidate solutions has been previously proposed [47], however never fully implemented. This method could potentially provide a viable alternative approach for ambiguity resolution, however it does not ensure that minimal intervals between solutions is found.

Comparison with Huang's Fast Ambiguity Resolution Method

Huang's Fast Ambiguity Resolution Method [60] uses hypothesis testing to construct an ambiguity acceptance space which is then converted to linear form using singular value decomposition. Particle swarm techniques are used to quickly find an initial solution. Huang provides an analysis of ambiguity resolution time vs problem size. A comparison of speed, complexity, accuracy should be performed in order to evaluate performance of the proposed algorithm. Thus, Huang's algorithm is a good marker of performance comparison and can be a useful tool in the development and testing of the algorithm architecture.

CHAPTER VII

SUMMARY AND CONCLUSIONS

This chapter contains reflection regarding the specific immediate results which have been achieved by the methods described in this dissertation in addition to the broader implications of the completed work. The described methods require further testing in real-world applications in order to understand their utility at the mission-level scope. Ultimately, even if these methods are technically successful, if they do not provide sufficient utility to engineers, operators, and managers, they will likely not be implemented in the future.

Summary

This section provide a recap of the methods used in OpNav and X-Nav and any results from testing and performance verification.

OpNav

OpNav is a means of autonomous optical navigation for spacecraft which uses visible light imagery of extended object planets or moons in order to estimate the position of the observing spacecraft with respect to the observed body. OpNav relies on four phases 1) image preprocessing and camera calibration, 2)initial processing and position estimate, 3) nonlinear iterative least squares limb fitting using a Gaussian on the image gradient for refined position estimate, 4) translation and rotation of position estimate into relevant coordinate frame with respect to desired reference body. The methods developed here were tested on real and synthetic data and successfully met the pointing requirements given by NASA when performing centroid and apparent size estimation of observed bodies. Algorithm can provide centroiding precision on the order of $\sigma = 0.12$ px when used on real images of the Moon taken from the ISS. This equates to a distance estimation error of about 500 km when performing OpNav at Earth-Moon distances.

X-Nav

X-Nav is a means of autonomous position estimation for spacecraft using the signals from distant X-ray pulsars. The method holds promise as a universal navigation solution during interplanetary and deep space missions and can potentially be a method of synchronizing ship clocks and estimating position, velocity, and attitude. The main difficulties of developing this technology lie in 1) developing X-ray sensor technology that can accurately process and resolve signals 2) resolving signal ambiguity and 3) compensating for effects of interstellar medium and gravitational disturbances. The ambiguity problem arises from the fact that pulsar signal are periodic and that an observer can potentially have multiple ambiguous candidate solutions proposed by X-Nav. The algorithm proposed in this paper aims to resolve this ambiguity problem by subselecting likely pulse wave intersection locations and then filtering out false candidate solutions. The method assumes perfect period knowledge, planar wavefront, and successful signal processing. It is developed using two spatial dimensions and one temporal dimensions but can be extended to n spatial dimensions. The proposed algorithm still requires implantation and testing.

Conclusion

This section aims to contextualize the work and results described in this dissertation. Speculation regarding proper application of the developed algorithms as well as commentary regarding the broader, far-reaching, utility of the methods is provided.

OpNav

The results of OpNav testing indicate that it is sufficiently accurate to serve as a backup navigation system for interplanetary and cislunar missions. However, it is likely not appropriate for navigation when near a large planet or moon. If the body can not be observed in its entirety, then the OpNav algorithms proposed here would not work. It may be possible that while in orbit around one planet or moon, that another planet or moon which appears as an extended body is used for OpNav. This is technically possible, as was done with image processing of Moon images taken from the ISS, however, the results would not be very accurate and it is likely not a useful capability under those operating conditions. The distance estimation error of about 500 km when using OpNav on board the ISS may be impressive considering the distance from the Earth to the Moon is around a thousand times greater than the error, however it's probably not sufficient precision for orbital maneuvering around Earth. It's likely that other forms of relative navigation are more useful in this case.

The OpNav algorithms presented here have been successfully tested at Johnson Space Center on synthetic and real image data. They have been undergoing conversion to embedded C code in order to be compatible with Orions flight systems.

Before they see flight heritage they must undergo integrated hardware testing where a simulated Orion will fly through simulated space and attempt to image bodies, perform image processing, navigate, and perform maneuvers successfully.

Though developed for Orion, these OpNav algorithms are vehicle agnostic. The only constraint which they face is that a separate system, other than the OpNav imaging system is needed in order to estimate attitude, which in turn is necessary to generate an OpNav position estimate. Future developments could look into using the imagery taken from OpNav to estimate attitude as well, and thus provide a fully independent navigation solution.

The effects of successful implementation of these algorithms should be reflected by increased safety margins, and reduced ground crew effort in navigation, and increased capabilities of Orion during loss-of-comms scenarios compared to current performance. To evaluate these broader goals, extensive system-level testing is needed which potentially would involve simulated crew operations as well. This analysis is necessary in order to determine the ultimate utility of OpNav as a backup, and potentially additional navigation system. Upon successful testing, OpNav can hopefully serve, not only as a contingency measure, but be run in parallel with other navigation systems and increase the capabilities of a spacecraft while on mission.

X-Nav

The proposed method of resolving ambiguity remains to be implemented and tested. Proper testing would likely require the use of real pulsar TOA data along with a simulation environment which can create synthetic TOA_{obs} values based on where an observer is located in space and time. There are already existing methods for resolving ambiguity, and thus a comparison of performance is warranted in the future. Even if the proposed method does not appear to perform as well as some existing methods, it can perhaps serve as a foundation for solving the minimal interval problem. While existing methods are able to resolve ambiguity, they do not seem to provide an estimate of the minimal solution interval: $[T_{\min}, D_{\min}]$. It remains to be seen if it is possible to use a algorithmic approach to solving this problem in a way that guarantees convergence and allows zero false positives while also not missing any potential minimal solution intervals. More work is required to develop methods of investigating this problem. If it can be solved, the implications would likely extend beyond the field of spacecraft navigation as a general tool for estimating distances between lattice points when there is uncertainty in the lattice point locations. These results could thus be relevant to the study of lattice algorithms which has applications in many fields of computer science such as cryptography and combinatorial optimization.

Future Work

This section contains potential ways of refining algorithms and verifying performance. All of the following recommendations regarding future work are either mathematical or algorithmic in nature. Integrated hardware testing, or systems level developments, is not considered.

OpNav – Refinement of $\theta(\xi)$ *Function*

The relationship between ξ and θ was modeled as a simple scaling as shown in Equation 3.14, however this linear approximation produced up to a 15% error in estimation of ξ as shown in Figure III.17. A numerical estimate of the derivative $d\xi/d\theta$ looks linear over the domain of interest, which points towards using a quadratic approximation to model ξ as a function of θ. This adds a minor complication, when inverting this relationship in order to estimate θ as a function of ξ is a little more complex. Also, the partial derivatives of the cost function with respect to ξ will change.

OpNav – Asymmetric Gaussian Limb Model

It can be noted in Figure III.23 that the graytone data associated with the limb of the Earth is nosier on the Earth side of the limb. In order to capture this asymmetry in the distribution of data around the limb, an asymmetric fitting function could be used. Initial tests have been done using fitting function made from the sum of a Gaussian and sigmoid function where the transition of the sigmoid is aligned with the center of the Gaussian.

OpNav – Super-Resolution Image Gradient

The gradient of the image used in OpNav is computed using numerical differencing techniques. Once the gradients are computed a new image is generated with the same resolution as the original image. However, it is not necessary for the new image to have the same resolution as the original, and it's possible that a much higher resolution image is generated to represent the gradient. Whether or not this would improve the results of limb fitting is unclear, but it would provide much more pixels to use for the least squares fit. It would also reduce discretization effects, which might improve fit quality.

OpNav – Estimating Uncertainty of Position Fix Using Pixel Graytone Statistics

The current formulation of the limb fitting in OpNav does not provide a confidence measure along with the position estimate. The confidence of the position estimate can have uncertainty introduced into it from several sources: timing noise, optical distortion, inaccuracy in ephemeris data, and image noise. Image noise would affect the quality of the limb fit. So the uncertainty associated with graytone level due to noise can be propagated into the uncertainty of the position estimate, and thus make it more useful for implementation into a navigation filter.

OpNav – Study of Atmospheric Bias on Limb Fitting

Atmospheres alter the appearance of the limb of planets and make fro a more gradual transition across the limb. The effect this has is to alter the apparent size of the observed planet, generally biasing the results towards a larger appearance of the target body and thus a closer distance estimate. This effect is only apparent when close to the target body, and reduces as one moves away. This exact relationship can be studied. Studies of this relationship have already been performed by Christian [61].

X-Nav – Pulse TOA Simulator

Attempting to improve the performance of the proposed ambiguity resolution method is not necessary until an initial implementation is completed. However, testing would benefit from building a pulsar simulator, or at least a simulator of pulse TOAs. Such a simulator would not have to handle the complexities of generating realistic photon streams and then processing the signal, but it might provide a useful tool for generating TOA values with realistic timing noise and periods.

X-Nav – Extension to 3D and Testing Against Other Pulse Ambiguity Resolvers

The development of the ambiguity resolution algorithm was performed in two dimensions, however, there is no reason why the method cannot readily be extended into three spatial dimensions. A few of the steps will change when this is done. First, wavefronts would be represented by planes instead of lines, and thus timing noise will create volumes of finite thickness. *Step 5* was originally a search for intersection of triplets of fronts in two dimensions. In three dimensions this will be a quadruplet intersection problem, searching for moments in time and locations in space where four wavefronts meet. Thus at the end of *Step 5* signals from the first four pulsars would be analyzed. *Step 6* will still analyzes conjunction between pairs of intersection points, however, each intersection point is now defined by three pulse waves, so this step will actually be processing pulse waves from six pulsars. Therefore, at *Step 7* the algorithm will be checking for conjunction checks starting with the first seven pulsars up through all N pulsars.

REFERENCES

[1] Franzen, A., Gillessen, S., Hermann, G., and Hinton, J., “Optical Observations of the Crab Pulsar using the first HESS Cherenkov Telescope,” *HESS contributions to the 28th International Cosmic Ray Conference Tsukuba, Japan*, 2003, p. 57.

[2] Mortari, D., “A fast on-board autonomous attitude determination system based on a new star-ID technique for a wide FOV star tracker,” *Advances in the Astronautical Sciences*, Vol. 93, 1996, pp. 893–904.

[3] Mortari, D., Samaan, M. A., Bruccoleri, C., and Junkins, J. L., “The pyramid star identification technique,” *Navigation*, Vol. 51, No. 3, 2004, pp. 171–183.

[4] Mortari, D., “Moon-Sun Attitude Sensor,” *Journal of spacecraft and rockets*, Vol. 34, No. 3, 1997, pp. 360–364.

[5] Samaan, M. A., Mortari, D., Pollock, T., and Junkins, J. L., “Predictive centroiding for single and multiple FOVs star trackers,” *Advances in the Astronautical Sciences*, Vol. 112, No. 979, 2002, pp. 59–71.

[6] Christian, J. A. and Robinson, S. B., “Noniterative horizon-based optical navigation by cholesky factorization,” *Journal of Guidance, Control, and Dynamics*, 2016, pp. 2757–2765.

[7] Zurek, R. W. and Smrekar, S. E., "An overview of the Mars Reconnaissance Orbiter (MRO) science mission," *Journal of Geophysical Research: Planets*, Vol. 112, No. E5, 2007.

[8] Jacobson, R., "The Orbits of the Regular Jovian Satellites," *European Planetary Science Congress*, Vol. 9, 2014.

[9] Owen Jr, W. M., Dumont, P. J., and Jackman, C. D., "Optical navigation preparations for New Horizons Pluto flyby," 2012.

[10] Fujiwara, A., Kawaguchi, J., Yeomans, D., Abe, M., Mukai, T., Okada, T., Saito, J., Yano, H., Yoshikawa, M., Scheeres, D., et al., "The rubble-pile asteroid Itokawa as observed by Hayabusa," *Science*, Vol. 312, No. 5778, 2006, pp. 1330–1334.

[11] Owen, W. M., "Methods of Optical Navigation," *AIAA/AAS Spaceflight Mechanics Conference, AAS 11-215,*, February 2011, pp. 1–19.

[12] Seidelmann, P. K. and Urban, S. E., "Explanatory Supplement to the Astronomical Almanac," 1992.

[13] Bernard, D., Doyle, R., Riedel, E., Rouquette, N., Wyatt, J., Lowry, M., and Nayak, P., "Autonomy and software technology on NASA's Deep Space One," *IEEE Intelligent Systems and their Applications*, Vol. 14, No. 3, 1999, pp. 10–15.

[14] Rayman, M. D., Varghese, P., Lehman, D. H., and Livesay, L. L., "Results from the Deep Space 1 technology validation mission," *Acta Astronautica*, Vol. 47, No. 2, 2000, pp. 475–487.

[15] Rayman, M. D., "The successful conclusion of the Deep Space 1 Mission: important results without a flashy title," *Space Technology*, Vol. 23, No. 2, 2003, pp. 185–196.

[16] Li, S., Xu, C., and Xie, M., "A robust O (n) solution to the perspective-n-point problem," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 34, No. 7, 2012, pp. 1444–1450.

[17] Wahba, G., "A least squares estimate of satellite attitude," *SIAM review*, Vol. 7, No. 3, 1965, pp. 409–409.

[18] Green, B., "Canny edge detection tutorial," *Retrieved: March*, Vol. 6, 2002, pp. 2005.

[19] Rong, W., Li, Z., Zhang, W., and Sun, L., "An improved CANNY edge detection algorithm," *2014 IEEE International Conference on Mechatronics and Automation*, IEEE, 2014, pp. 577–582.

[20] Derpanis, K. G., "The harris corner detector," *York University*, 2004, pp. 1–2.

[21] Ryu, J.-B., Lee, C.-G., and Park, H.-H., "Formula for Harris corner detector," *Electronics letters*, Vol. 47, No. 3, 2011, pp. 180–181.

[22] Pedersen, S. J. K., "Circular hough transform," *Aalborg University, Vision, Graphics, and Interactive Systems*, Vol. 123, No. 6, 2007.

[23] Rizon, M., Haniza, Y., Puteh, S., Yeon, A., Shakaff, M., Abdul Rahman, S., Sugisaka, M., Sazali, Y., M Rozailan, M., and Karthigayan, M., "Object detection using circular Hough transform," 2005.

[24] Bradski, G. and Kaehler, A., *Learning OpenCV: Computer vision with the OpenCV library*, " O'Reilly Media, Inc.", 2008.

[25] Weisstein, E. W., "Gnomonic projection," 2001.

[26] Shuster, M. D. et al., "A survey of attitude representations," *Navigation*, Vol. 8, No. 9, 1993, pp. 439–517.

[27] Markley, F. L., "Attitude determination using vector observations and the singular value decomposition," *Journal of the Astronautical Sciences*, Vol. 36, No. 3, 1988, pp. 245–258.

[28] Mortari, D., "ESOQ-2 single-point algorithm for fast optimal spacecraft attitude determination," *Advances in the Astronautical Sciences*, Vol. 95, 1997, pp. 817–826.

[29] Markley, F. L. and Mortari, D., "Quaternion attitude estimation using vector observations." *Journal of the Astronautical Sciences*, Vol. 48, No. 2, 2000, pp. 359–380.

[30] Mortari, D., De Dilectis, F., and Zanetti, R., "Position Estimation using the Image Derivative," *Aerospace*, Vol. 2, No. 3, 2015, pp. 435–460.

[31] Mortari, D., DSouza, C. N., and Zanetti, R., "Image processing of illuminated ellipsoid," *Journal of Spacecraft and Rockets*, 2016, pp. 448–456.

[32] Borissov, S. and Mortari, D., "Pose Estimation using Optical Camera in Lunar Orbit," *2014 AAS/AIAA Space Flight Mechanics Meeting Conference, AAS 14-247*, January 2014.

[33] Borissov, S. and Mortari, D., "Optimal Single-Point and Filtered Pose Estimation for Lunar Orbiters Using Visible Camera," *Proc. 24th AAS/AIAA Space Flight Mechan. Meet.*, 2014.

[34] de Dilectis, F., "Vision-based Autonomous Navigation using Moon and Earth images," *Doctoral dissertation, Texas A & M University*, 2014.

[35] Brisebarre, N. and Joldeş, M., "Chebyshev interpolation polynomial-based tools for rigorous computing," *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, 2010, pp. 147–154.

[36] Zhang, Z., "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, December 2000, pp. 1330–1334, MSR-TR-98-71, Updated March 25, 1999.

[37] Gonzalez, R. C., Woods, R. E., and Eddins, S. L., *Digital image processing using MATLAB*, Pearson Education India, 2004.

[38] Fischler, M. A. and Bolles, R. C., "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.

[39] Christian, J. A., "Optical navigation using iterative horizon reprojection," *Journal of Guidance, Control, and Dynamics*, 2016, pp. 1092–1103.

[40] Borissov, S. and Mortari, D., "Centroiding and Sizing Optimization of Ellipsoid Image Processing using Nonlinear Least-Squares," *2018 AAS/AIAA Astrodynamics Specialist Conference, AAS 18-229*, August 2018.

[41] Borissov, S. and Mortari, D., "Image Processing of Earth and Moon Images for Optical Navigation Systems," *2015 AAS/AIAA Astrodynamics Specialist Conference, AAS 15-744*, August 2015.

[42] Winternitz, L. M., Mitchell, J. W., Hassouneh, M. A., Valdez, J. E., Price, S. R., Semper, S. R., Wayne, H. Y., Ray, P. S., Wood, K. S., Arzoumanian, Z., et al., "SEXTANT X-ray Pulsar Navigation demonstration: Flight system and test results," *2016 IEEE Aerospace Conference*, IEEE, 2016, pp. 1–11.

[43] Zhang, X., Shuai, P., Huang, L., Chen, S., and Xu, L., "Mission overview and initial observation results of the X-ray pulsar navigation-I satellite," *International Journal of Aerospace Engineering*, Vol. 2017, 2017.

[44] Richards, G., "Project Daedalus-The navigation problem," *Journal of the British Interplanetary Society*, Vol. 28, 1975, pp. 150–160.

[45] Bertaux, J.-L. and Lallement, R., "On the Stability of the Interstellar Wind through the Solar System," *Journal of Physics: Conference Series*, Vol. 577, IOP Publishing, 2015, p. 012004.

[46] Heerikhuisen, J. and Pogorelov, N., "An estimate of the nearby interstellar magnetic field using neutral atoms," *The Astrophysical Journal*, Vol. 738, No. 1, 2011, pp. 29.

[47] McConnell, S., McCarthy, M., Buchanan, T., Tang, K., Wasson, M., Young, T., Borissov, S., and Mortari, D., "X-Ray Pulsar Navigation Position Deter-

mination using Spherical Volumes," *2018 AAS/AIAA Astrodynamics Specialist Conference, AAS 18-354*, August 2018.

[48] Winternitz, L. M., Hassouneh, M. A., Mitchell, J. W., Valdez, J. E., Price, S. R., Semper, S. R., Wayne, H. Y., Ray, P. S., Wood, K. S., Arzoumanian, Z., et al., "X-ray pulsar navigation algorithms and testbed for SEXTANT," *2015 IEEE Aerospace Conference*, IEEE, 2015, pp. 1–14.

[49] Becker, W., Bernhardt, M. G., and Jessner, A., "Autonomous spacecraft navigation with pulsars," *arXiv preprint arXiv:1305.4842*, 2013.

[50] Sala Álvarez, J., Urruela Planas, A., Villares Piera, N. J., Estalella, R., and Paredes, J. M., "Feasibility study for a spacecraft navigation system relying on pulsar timing information," 2004.

[51] Sheikh, S. I., Pines, D. J., Ray, P. S., Wood, K. S., Lovellette, M. N., and Wolff, M. T., "Spacecraft Navigation using X-ray pulsars," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, 2006, pp. 49–63.

[52] Wood, K. S., "Navigation studies utilizing the NRL-801 experiment and the ARGOS satellite," *Small Satellite Technology and Applications III*, Vol. 1940, International Society for Optics and Photonics, 1993, pp. 105–116.

[53] HUANG, Z., LI, M., and SHUAI, P., "Study on Methods of Ambiguity Resolution for the X-Ray Pulsar Navigation System [J]," *Aerospace Control and Application*, Vol. 1, 2010.

[54] Li, X., Hao, H., Lei, G., and Shen, S., "A quick method of phase ambiguity resolution with X-ray pulsar navigation," *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 4, No. 18, 2012, pp. 3432–3436.

[55] Feng, D.-j., Xu, L.-p., Zhang, H., and Song, S.-b., "Determination of inter-satellite relative position using X-ray pulsars," *Journal of Zhejiang University SCIENCE C*, Vol. 14, No. 2, 2013, pp. 133–142.

[56] Arumugam, S. and Jermaine, C., "Closest-point-of-approach join for moving object histories," *22nd International Conference on Data Engineering (ICDE'06)*, IEEE, 2006, pp. 86–86.

[57] Kannan, R., "Minkowski's convex body theorem and integer programming," *Mathematics of operations research*, Vol. 12, No. 3, 1987, pp. 415–440.

[58] Gruber, P. M. and Lekkerkerker, C. G., *Geometry of numbers*, North-Holland, 1987.

[59] Minkowski, H., *Geometrie der zahlen*, Vol. 2, , 2016.

[60] Huang, L., Lin, Q., Zhang, X., and Shuai, P., "Fast ambiguity resolution for pulsar-based navigation by means of hypothesis testing," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 53, No. 1, 2017, pp. 137–147.

[61] Christian, J. A., "Horizon-Based Optical Navigation Using Images of a Planet with an Atmosphere," *AIAA/AAS Astrodynamics Specialist Conference*, 2016, p. 5442.

APPENDIX A

PARTIAL DERIVATIVES OF ξ AND M

ξ Derivatives

$$\frac{\partial \xi_j}{\partial x} = \boldsymbol{b}_j^{\mathrm{T}} \frac{\partial M_B}{\partial x}\,\boldsymbol{b}_j \qquad \frac{\partial \xi_j}{\partial y}\boldsymbol{b}_j^{\mathrm{T}} \frac{\partial M_B}{\partial y}\,\boldsymbol{b}_j \qquad \frac{\partial \xi_j}{\partial z}\boldsymbol{b}_j^{\mathrm{T}} \frac{\partial M_B}{\partial z}\,\boldsymbol{b}_j \tag{A.1}$$

M Derivatives

M_B-matrix and its partials (triaxial body)

The M_B matrix for a triaxial body has the expression

$$M_B = \frac{1}{a^2b^2c^2}\begin{bmatrix} b^2c^2 - c^2y^2 - b^2z^2 & c^2xy & b^2xz \\ c^2xy & a^2c^2 - a^2z^2 - c^2x^2 & a^2yz \\ b^2xz & a^2yz & a^2b^2 - b^2x^2 - a^2y^2 \end{bmatrix} \tag{A.2}$$

whose partials are

$$\frac{\partial M_B}{\partial x} = \frac{1}{a^2b^2c^2}\begin{bmatrix} 0 & c^2y & b^2z \\ c^2y & -2c^2x & 0 \\ b^2z & 0 & -2b^2x \end{bmatrix}, \quad \frac{\partial M_B}{\partial y} = \frac{1}{a^2b^2c^2}\begin{bmatrix} -2c^2y & c^2y & 0 \\ c^2y & 0 & a^2z \\ 0 & a^2z & -2a^2y \end{bmatrix}, \tag{A.3}$$

and

$$\frac{\partial M_B}{\partial z} = \frac{1}{a^2b^2c^2}\begin{bmatrix} -2b^2z & 0 & b^2x \\ 0 & -2a^2z & a^2y \\ b^2x & a^2y & 0 \end{bmatrix}. \tag{A.4}$$

M_B*-matrix and its partials (Earth)*

The Earth can be considered an axial symmetric ellipsoid with two identical equatorial semi-axes, $a = b = R_E$, and a polar semi-axis, $c = R_P$.

$$M_B = \frac{1}{a^4c^2}\begin{bmatrix} a^2(c^2-z^2)-c^2y^2 & c^2xy & a^2xz \\ c^2xy & a^2(c^2-z^2)-c^2x^2 & a^2yz \\ a^2xz & a^2yz & a^2(a^2-x^2-y^2) \end{bmatrix} \tag{A.5}$$

whose partials are

$$\frac{\partial M_B}{\partial x} = \frac{1}{a^4c^2}\begin{bmatrix} 0 & c^2y & a^2z \\ c^2y & -2c^2x & 0 \\ a^2z & 0 & -2a^2x \end{bmatrix}, \quad \frac{\partial M_B}{\partial y} = \frac{1}{a^4c^2}\begin{bmatrix} -2c^2y & c^2y & 0 \\ c^2y & 0 & a^2z \\ 0 & a^2z & -2a^2y \end{bmatrix}, \tag{A.6}$$

and

$$\frac{\partial M_B}{\partial z} = \frac{1}{a^4c^2}\begin{bmatrix} -2a^2z & 0 & a^2x \\ 0 & -2a^2z & a^2y \\ a^2x & a^2y & 0 \end{bmatrix}. \tag{A.7}$$

M_B*-matrix and its partials (Earth, scaled)*

Previous matrices can be scaled by setting $\rho = \dfrac{R_E}{R_P} = \dfrac{a}{c} > 1$,

$$M_B = \begin{bmatrix} 1-y^2-\rho^2z^2 & xy & \rho^2xz \\ xy & 1-x^2-\rho^2z^2 & \rho^2yz \\ \rho^2xz & \rho^2yz & \rho^2(1-x^2-y^2) \end{bmatrix} \tag{A.8}$$

whose derivatives are

$$\frac{\partial M_B}{\partial x} = \begin{bmatrix} 0 & y & \rho^2 z \\ y & -2x & 0 \\ \rho^2 z & 0 & -2\rho^2 x \end{bmatrix}, \quad \frac{\partial M_B}{\partial y} = \begin{bmatrix} -2y & x & 0 \\ x & 0 & \rho^2 z \\ 0 & \rho^2 z & -2\rho^2 y \end{bmatrix}, \tag{A.9}$$

and

$$\frac{\partial M_B}{\partial z} = \begin{bmatrix} -2\rho^2 z & 0 & \rho^2 x \\ 0 & -2\rho^2 z & \rho^2 y \\ \rho^2 x & \rho^2 y & 0 \end{bmatrix}. \tag{A.10}$$

M_B-matrix and its partials (Moon)

Moon can be consider a sphere, $a = b = c$. Therefore,

$$M_B = \frac{1}{a^4} \begin{bmatrix} a^2 - y^2 - z^2 & xy & xz \\ xy & a^2 - z^2 - x^2 & yz \\ xz & yz & a^2 - x^2 - y^2 \end{bmatrix} \tag{A.11}$$

with derivatives

$$\frac{\partial M_B}{\partial x} = \frac{1}{a^4} \begin{bmatrix} 0 & y & z \\ y & -2x & 0 \\ z & 0 & -2x \end{bmatrix}, \quad \frac{\partial M_B}{\partial y} = \frac{1}{a^4} \begin{bmatrix} -2y & y & 0 \\ y & 0 & z \\ 0 & z & -2y \end{bmatrix} \tag{A.12}$$

and

$$\frac{\partial M_B}{\partial z} = \frac{1}{a^4} \begin{bmatrix} -2z & 0 & x \\ 0 & -2z & y \\ x & y & 0 \end{bmatrix} \tag{A.13}$$

M_B***-matrix and its partials (Moon, scaled)***

$$a = b = c \text{ and } \rho = \frac{R_E}{R_P} = 1$$

$$M_B = \begin{bmatrix} 1 - y^2 - z^2 & xy & xz \\ xy & 1 - x^2 - z^2 & yz \\ xz & yz & 1 - x^2 - y^2 \end{bmatrix} \tag{A.14}$$

with derivatives

$$\frac{\partial M_B}{\partial x} = \begin{bmatrix} 0 & y & z \\ y & -2x & 0 \\ z & 0 & -2x \end{bmatrix}, \quad \frac{\partial M_B}{\partial y} = \begin{bmatrix} -2y & x & 0 \\ x & 0 & z \\ 0 & z & -2y \end{bmatrix} \tag{A.15}$$

and

$$\frac{\partial M_B}{\partial z} = \begin{bmatrix} -2z & 0 & x \\ 0 & -2z & y \\ x & y & 0 \end{bmatrix} \tag{A.16}$$