EMBRY-RIDDLE
Aeronautical University
SCHOLARLY COMMONS

Dissertations and Theses

5-2019

Space Image Processing and Orbit Estimation Using Small Aperture Optical Systems

David Zuehlke

Follow this and additional works at: https://commons.erau.edu/edt

Part of the Aerospace Engineering Commons

Scholarly Commons Citation
Zuehlke, David, "Space Image Processing and Orbit Estimation Using Small Aperture Optical Systems" (2019). *Dissertations and Theses*. 460.
https://commons.erau.edu/edt/460

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

SPACE IMAGE PROCESSING AND ORBIT ESTIMATION USING SMALL APERTURE OPTICAL SYSTEMS

A Thesis

Submitted to the Faculty

of

Embry-Riddle Aeronautical University

by

David Zuehlke

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aerospace Engineering

May 2019

Embry-Riddle Aeronautical University

Daytona Beach, Florida

SPACE IMAGE PROCESSING AND ORBIT ESTIMATION
USING SMALL APERTURE OPTICAL SYSTEMS

by

David Zuehlke

A Thesis prepared under the direction of the candidate's committee chairman, Dr. Troy Henderson, Department of Aerospace Engineering, and has been approved by the members of the thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chairman, Dr. Troy Henderson

Member, Dr. Richard Prazenica

Morad Nazari

Member, Dr. Morad Nazari

J Alan Lovell

Member, Dr. Alan Lovell

Dr. Magdy Attia
Graduate Program Coordinator

4. 24. 2019
Date

Dr. Maj Mirmirani
Dean of College of Engineering

4/24/2019
Date

Dr. Lon Moeller
Senior Vice President for Academic Affairs and Provost

4/24/19
Date

ACKNOWLEDGMENTS

The heavens declare the glory of God, and the sky above proclaims His handiwork. Day to day pours out speech, and night to night reveals knowledge.
Psalm 19:1-2

Anything worth doing, is worth doing well, but nothing worth doing can be done alone. The author would like to acknowledge the following individuals and organizations without whom this work would never have been completed. Thank you to my wonderful family, without who's support I would never have been able to finish. Especially to my mom, her admonition to never do less than my best and that "Math is fun" inspired me to reach for new heights. To all my friends who helped me on late nights of research, thank you. To Madhur Tiwari, thank you for reminding me of the ever important coffee break. Thank you to my committee members, especially Dr. Henderson, whose advice and direction has been invaluable. And finally, thank you to the Florida Space Grant Consortium by which this work was partially funded.

Soli Deo Gloria!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

SYMBOLS

A	Matrices are non-bold capital font
a_i	Scalar values are non-bold *italic* font
$\mathbf{r_i}$	Vectors are bold font
$\mathbf{F}$	Force
G	Universal gravitational constant
m	mass
$\mathbf{r_i}$	Satellite position vector (bold font used for vectors)
$\mathbf{v_i}$	Satellite velocity vector (subscript indicates time)
$\ddot{\mathbf{r}}$	Satellite acceleration vector
$\mathbf{R_i}$	Site position vector
μ	Gravitational constant
μ_e	earth's gravitational constant $= 3.9864 \times 10^5 km^2/s^3$
$\widehat{\mathbf{L}}_\mathbf{i}$	Line of sight unit vector from observer to target
α_t	Topocentric Right Ascension
δ_t	Topocentric Declination
a	Semimajor axis
e	Eccentricity
i	Inclination
Ω	Right Ascension of the Ascending Node
ω	Argument of Perigee
θ	True anomaly, gradient orientation
f	True anomaly (Schaub's notation)
f and g	Lagrange Gibbs, f and g solutions
M	Mean anomaly
E	Eccentric anomaly
τ	Modified time variable
λ_j	Gooding's notation for LOS unit vector
t_i	Observation time

ω_{earth}	Rotation rate of earth
ρ	Unknown range to an observed LOS
$\{\delta_x, \delta_y\}$	Newton-Raphson derivatives
J	N-Gooding cost function cost
σ	Standard deviation
P	Covariance matrix
χ_0	Sigma points (UT)
$I(x, y)$	Image intensity matrix
$G(x, y, \sigma)$	Gaussian smoothing Kernel
$*$	Convolution operation
K_x, K_y	Prewwit operators
K^T	Matrix transpose operation
$\nabla I(x, y)$	Image gradient
$Z(x, y)$	Image tensor matrix
λ	Eigenvalues
$c_h(x, y)$	Cornerness metric
g_{mag}	Gradient magnitude
g_x	Image gradient in x-direction
g_y	Image gradient in y-direction
$GmagI(x, y)$	Image gradient magnitude
$\mathbf{x_c}$	Centroid (x, y) coordinates
$\mathbf{v_{cam}}$	Camera frame vector
R_{cam}	Camera rotation matrix
Ψ_{obs}	“Observability” metric

ABBREVIATIONS

AIOD	Angles-Only Initial Orbit Determination
SSA	Space Situational Awareness
RSO	Resident Space Object
IOD	Initial Orbit Determination
POD	Precise Orbit Determination
UT	Unscented Transform
OD	Orbit Determination
KF	Kalman Filter
TLE	Two-Line Element
ECI	Earth-Centered Inertial Coordinate Frame
LOS	Line of Sight Vector
PSF	Point Spread Function
SGP4	Special General Perturbations 4

NOMENCLATURE

Astrodynamics	Celestial mechanics applied to artificial satellites.
Image Plane	Pixel coordinate frame of a camera image.
Satellite State Vector	Elements describing the current state of a satellite.
Unperturbed Keplerian Motion	Orbital motion with only gravity as a force.
Orbital Elements	$\{a, e, i, \Omega, \omega, \theta\}$.

ABSTRACT

Zuehlke, David MSAE, Embry-Riddle Aeronautical University, May 2019. Space Image Processing and Orbit Estimation Using Small Aperture Optical Systems.

Angles-only initial orbit determination (AIOD) methods have been used to find the orbit of satellites since the beginning of the Space Race. Given the ever increasing number of objects in orbit today, the need for accurate space situational awareness (SSA) data has never been greater. Small aperture $(< 0.5m)$ optical systems, increasingly popular in both amateur and professional circles, provide an inexpensive source of such data. However, utilizing these types of systems requires understanding their limits. This research uses a combination of image processing techniques and orbit estimation algorithms to evaluate the limits and improve the resulting orbit solution obtained using small aperture systems. Characterization of noise from physical, electronic, and digital sources leads to a better understanding of reducing noise in the images used to provide the best solution possible. Given multiple measurements, choosing the best images for use is a non-trivial process and often results in trying all combinations. In an effort to help autonomize the process, a novel ”observability metric” using only information from the captured images was shown empirically as a method of choosing the best observations. A method of identifying resident space objects (RSOs) in a single image using a gradient based search algorithm was developed and tested on actual space imagery captured with a small aperture optical system. The algorithm was shown to correctly identify candidate RSOs in a variety of observational scenarios.

1. Introduction

The prospect of new mega–constellations such as OneWeb, and SpaceX's Starlink constellation beg for a solution to providing real time space surveillance around the globe. However, existing space situational awareness (SSA) infrastructure and facilities are not equipped to efficiently handle the influx of objects to track. Low cost, commercially available telescopes provide a lucrative option for providing useful SSA data. DARPA's Orbit Outlook program seeks to use data from a network of technically inclined amateur astronomers instead of the monolithic tracking stations that are part of the Air Force Space Surveillance Network (DARPA, n.d.). The notion can become a reality only if the capabilities and limitations of the telescopes that such amateurs use are well understood.

1.1 Problem Statement

Small aperture telescope systems are of great interest for the future of space surveillance. However, to maximize such system's utility for SSA, the capabilities and limitations of such systems must be well understood. To date, a comprehensive study of the entire end-to-end process of using such a system for angles-only orbit determination (AIOD) has not been performed. This research aims to meet this need by using image processing algorithms and orbit estimation techniques to test the limits of a small aperture optical system.

1.2 Motivation

No other time in history has seen such an explosive increase in the number of satellites in orbit. The planned constellations by SpaceX and OneWeb to place hundreds, or thousands, of satellites in low-earth-orbit present obvious challenges to current space surveillance capabilities. Cost effective solutions are needed to provide the needed data.

Small aperture optical telescopes for SSA present a viable option, however, currently there has not been an effort to combine all the necessary parts from gathering data to outputting an orbit estimate into a cohesive whole. This work attempts to provide a framework for the AIOD process using small aperture telescopes while introducing image processing and image discrimination algorithms to improve the AIOD solution.

1.3 Summary and Contribution of Work

This research seeks to characterize the end-to-end capability of performing AIOD using small aperture telescopes. Image processing and noise reduction algorithms are used to improve the orbit solution from angles-only data. A method of finding objects and discriminating Resident Space Objects (RSOs) from a single image is shown to correctly identify RSOs from several experimental image sets captured by the author. Once candidate RSOs have been identified, the orbit determination process is run on a set of images. The image set used for orbit estimation is chosen by using a new metric developed to help decide on the measurement images to use in order to produce the best possible orbit solution. A short list of the main contributions of this research is presented below.

Contributions of Work

- A method of identifying candidate RSOs given a single image using a gradient based search method.
- The development of an “observability” metric, Ψ_{obs}, to aid in autonomously choosing the best set of measurements to use in classical AIOD methods.
- Provide an in depth description of the end-to-end process from image capture to orbit estimate using a small aperture optical system.

- Apply noise reduction methods from astronomy to improve initial orbit estimates from experimental data.

1.4 Outline of Thesis

Chapter 2 gives a summary of the background information necessary for orbit determination in general and angles-only orbit determination in particular. Classical and modern orbit determination techniques are reviewed with an algorithmic pseudo-code given for key methods. A discussion of the different sources of observation noise as well as observation modes is presented. Chapter 3 contains the theoretical developments for this research. The image processing techniques used in this work are presented including noise characterization, noise reduction, and image calibration. A method of discriminating and labeling objects as stars, or satellites given a single image is presented. The theoretical basis for an "observability" metric for choosing the best observations from a data set is developed. To test the new metric and provide a baseline comparison for experimental results, a simple simulation tool for generating AIOD scenarios is outlined. Chapter 4 gives the results of applying the image processing techniques derived in chapter 3 to real observational data. Results are compared to simulated scenarios. Lastly, chapter 5 discusses important conclusions and future work based on the results presented in chapter 4.

2. Background and History

This chapter is broken into two main parts of literature review. First a section on orbital mechanics and the orbit determination problem is provided. Second, a section detailing the theory and history behind the optical orbit determination process is given, including discussions on the sources of noise for optical observations.

2.1 Orbital Mechanics

Before beginning a discussion of the history of any subject, a fitting definition of the subject must first be provided. Orbital mechanics (often referred to as astrodynamics) refers to applying physics and celestial mechanics to describe the motion of artificial satellites. The term orbital mechanics can also be used in a more general sense for the study of the orbiting motion of a general body such as a planet or asteroid. Using orbital mechanics, the orbit of any arbitrary body can be found if enough information is provided. Using the mathematics of orbital mechanics and measurements of an orbiting body in order to find the unknown orbit is called orbit determination.

2.1.1 A Brief History

The study of orbit determination rests very firmly on the shoulders of many of the greatest mathematical minds in history. The task of determining the orbit of a satellite begins with a thorough understanding of the physics behind orbital motion. Long before man dreamed of placing objects in space, mathematicians were greatly intrigued by the motion of the planets and other solar system bodies. From the earliest era of human history, discovering a pattern to the motion of heavenly bodies was the paramount task of mathematicians and astronomers.

Ptolemy, the great Greco-Roman astronomer and mathematician, published his famous astronomical treatise, the *Almagest* circa A.D. 100. Contained therein were convenient tables for predicting the positions of planets from a model he derived from observational data. Fast forward more than 13 centuries, and the science of predicting planetary positions had not changed much. An important development came though with Nicholas Copernicus positing his Heliocentric (sun-centered) universe. With this revelation, astronomers sought alternative explanations for planetary motions to the complex mathematical forms and epicycles used in the past.

Tycho Brahe (1546-1601), a Danish nobleman by birth and excellent observational astronomer by practice, made detailed observations of the positions of the stars and planets. Johannes Kepler, once Brahe's assistant, used Brahe's cataloged data to empirically show that the motion of the planets could be explained by elliptical motion around the sun. The death of Tycho Brahe in 1601 left Kepler with a vast store of observational data gathered under the late astronomer's watch. Kepler painstakingly examined these observational data, particularly observations of the planet Mars, seeking to find a mathematical model for the motion of the planets. In 1609, Kepler published his first two laws of planetary motion, but it took another decade until 1619 for him to formalize his third law of planetary motion and complete the set (Escobal, 1976).

Kepler's Laws

> Law I: Within the domain of the solar system all planets follow elliptical paths with the Sun at one focus.
>
> Law II: The radius vector from the Sun to a planet sweeps out equal areas in equal times.

> Law III: The squares of the periods of revolution of the planets about the Sun are proportional to the cubes of their mean distances from the Sun.

Kepler's first law was based on an empirical data fit from detailed observations of the motion of Mars. Kepler tried several different curves to approximate the motion of Mars, before finally settling on the ellipse. Seeing how well the data corresponded to elliptical motion he concluded that not just Mars, but all the planets in the solar system revolved around the sun on elliptical orbits.

In 1687, Sir Isaac Newton (1643-1727) provided the physical reasoning behind the motion of the planets. Newton's three laws, as well as his law of universal gravitation, form the basis of celestial mechanics. No study of orbital mechanics would be complete without Newton's Laws, for it is with these powerful tools that we derive the most basic equations of orbital mechanics.

Newton's Laws

> Law I: Every object persists in its state of rest or uniform motion in a straight line unless it is compelled to change that state by forces impressed on it.
>
> Law II: Force is equal to the change in momentum (mv) per change in time. For a constant mass, force equals mass times acceleration.
>
> Law III: For every action, there is an equal and opposite reaction ("Newton's Laws of Motion: Glenn Research Center", 2019).

The application of Newton's second law will help us develop the fundamental equations of orbital mechanics, while Newton's law of universal gravitation gives us an analytical expression for the force of gravity felt by all bodies in orbital motion. Newton's law of universal gravitation states that the force between any two particles is directly proportional to the product of their masses and inversely proportional to the square of the distance between the particles. Mathematically we have:

$$\mathbf{F} = -G\frac{m_1 m_2}{r^2}\hat{\mathbf{r}}. \tag{2.1}$$

Deriving the equations of unperturbed (gravity as the only active force) Keplerian motion comes from a straightforward application of Newton's second law and the Law of Universal Gravitation. Doing so allows us to derive the two-body motion mathematical model of orbits.

2.1.2 Two-Body Motion

Two-body motion, or two-body mechanics refers to the mathematics of orbital motion that follows unperturbed Keplerian motion. The physical conditions for two-body motion are given by Kepler's first two laws; elliptical motion about a central body, and equal areas swept out by an orbit in equal time (Vallado & McClain, 1997). "Unperturbed" refers to assuming that gravity is the only active force between the two bodies in question.

2.1.3 Two-Body Assumptions

Before we can use two-body mechanics for optical orbit determination, several important assumptions must be discussed. These assumptions are necessary for applying Newton's laws to the two-body dynamical system model. In the context of earth orbiting satellites, we will take one body to be the earth, and the other to be the orbiting satellite. An illustrative figure of such a system can be seen in Figure 2.1. There are four main

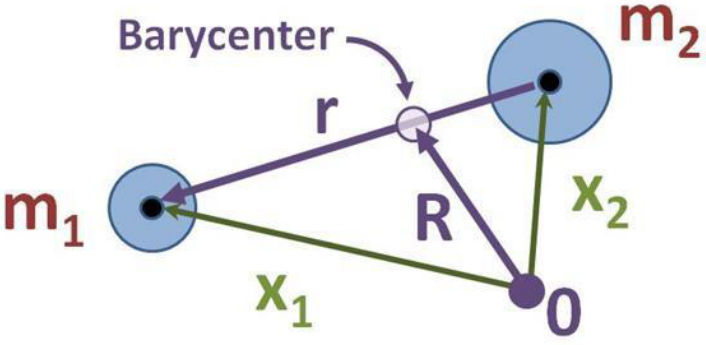


Figure 2.1. Two-Body Problem Setup

assumptions that must be followed to derive the equations of two-body mechanics. Given two masses, m_1, m_2, and Newton's laws we must assume the following.

1. $m_2 >> m_1$ (The earth is much more massive than any satellite).
2. Masses exhibit a spherically symmetric density distribution (i.e. each mass can be reduced to a point mass).
3. Gravity is the only force acting between the two bodies (i.e. ignore all perturbative forces).
4. An inertial coordinate system is in use (allows the use of regular vector math).

The first assumption follows from the fact that the mass of any orbiting satellite is so small in comparison to the earth that the gravitational force on the earth from the satellite can be safely ignored. The second assumption allows us to apply Newton's laws to the two-body system by requiring that the two bodies in consideration be modeled as point masses. The third assumption assumes that gravity is the dominant force acting between the two bodies, and that all other forces are small enough in magnitude or slow enough over the time of application to be ignored. The final assumption states that we must work in an inertial coordinate frame, which allows us to do regular vector calculus operations.

With these assumptions in mind, the derivation of the defining vector equation for describing a satellite is straightforward from an application of Newton's second law to the bodies while utilizing the law of universal gravitation for the force term. As such we obtain Equation 2.2, the fundamental differential equation for unperturbed two-body motion.

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \tag{2.2}$$

Equation 2.2 can be used to model the motion of any satellite (Schaub & Junkins, 2003). Note that this is the mathematical model for satellite motion utilized by all of the classical orbit determination methods.

2.1.4 Two-Body Motion Justification

The question might arise as to why unperturbed two-body motion is the mathematical model used for angles-only orbit determination. Oftentimes in engineering, approximations are made if they are are "close enough" to the truth. Such is the case with two-body mechanics for orbit determination. Given a typical angles-only orbit determination scenario (see section 2.2), only a fraction of the orbital arc of a target satellite is observed. Over this short arc of an orbit, the effects of perturbations are present, but are small enough to be safely ignored (Prussing & Conway, 1993). Furthermore, over the short arc observed, the effect of slow moving perturbations such as those due to earth's oblateness (J2) and solar radiation pressure (SRP) can be ignored. Prussing and Conway give a more detailed justification of two-body mechanics and assumptions for the interested reader (Prussing & Conway, 1993).

2.2 Angles-Only Orbit Determination

This section reviews the concept of angles-only orbit determination (AIOD), the process of calculating an initial orbit estimate of an unknown body using only angular measurements. In the context of this research, the unknown body will be assumed to be an earth orbiting artificial satellite. Optical (i.e. images captured through a telescope) data for orbit estimation are one source of the angular observations for AIOD. Generally a highly non-linear process, AIOD methods must use extra observations or special processing compared to other orbit determination (OD) methods to overcome the absence of range data in the observations.

To illustrate the fundamental shortcoming of optical data, consider Figure 2.2, which shows an image of a satellite taken by the author in 2019. Here we have an image of a satellite streaking through a background with stars. The individual pixels do not contain any information about the range to the unknown satellite. Furthermore, there is no direct relationship to the physical distance between the other objects in the image such as the background stars.

However, Figure 2.2 does contain enough information to uniquely determine two pieces of information. Using the known position of the background stars, or *a priori* pointing information, we can determine two angular measurements from the observation site to the observed satellite. These angles can be azimuth and elevation angles, or spherical coordinate angles such as right ascension (α_i) and declination (δ_i). Because a single image provides at most two angles, and since there are six state variables for a satellite, three images with two angular observations each are needed to uniquely determine a satellite's state.

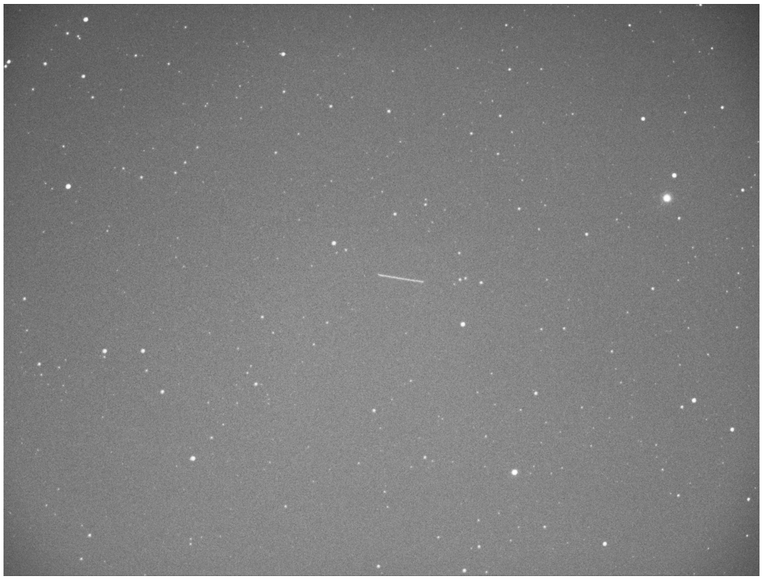


Figure 2.2. Example Image of Satellite

2.3 Ground-Based Vs. Space-Based Observations

The point of observation for orbit determination methods is, as one might expect very important. The problem of developing an orbit estimate of a satellite from a space-based observer is known as relative orbit estimation. This method is so named because the motion observed between measurements of the target satellite is motion relative to that of the observer satellite. However, it can be shown that the point of observation, though important, does not fundamentally change the AIOD process. The main difference lies in how the observer's position vector is updated between measurements. Also of importance will be observability considerations for both relative and ground-based orbit estimation. (Note, observability here refers to the physical geometric constraints and not the observability of control theory.) If the observer lies in the same orbital plane as the target, then there will be a fundamental indeterminancy in the observational data. For the space-based observer, this occurs when the observer satellite is in the same orbit as the target satellite.

2.3.1 Ground-Based Observations

For ground-based measurements, it is assumed that that the observer has a known position on the earth given by latitude, longitude, along with an observation time. These three pieces of information are enough to develop the position vector of the site at any specified time by transforming from the latitude, longitude coordinate system to the Earth Centered Inertial (ECI) system. For a comprehensive review of astrodynamic coordinate systems, please see chapter two of Vallado's Fundamental's of Astrodynamics (Vallado & McClain, 1997). When determining site vectors, the level of detail required depends greatly on the application. If all that is required is a quick approximation, then assuming the earth is spherical provides accurate enough results. However, it is not difficult to use the more accurate approximation model that accounts for the slightly ellipsoidal shape of the earth (Vallado & McClain, 1997). If an accurate site position vector is available without conversion from latitude and longitude, this would help reduce error in the final orbit estimate.

2.3.2 Space-Based Observations

When using space-based observations of satellites, the observer satellite's position is assumed to be known. The position can be known given a set of position vectors from onboard navigation, or from satellite catalog information in the form of Two-Line Element (TLE) data. The problem of running an AIOD method from the relative standpoint becomes a problem of properly attaining the observer's position at the time of each observation. It should be noted that Gooding's algorithm (section 2.7.2) is invariant to whether the observer resides in space or on the ground. Space-based observations have application to space surveillance, orbital debris identification, and proximity operations.

2.4 Satellite Observation Modes

The optical (image) data used to run the AIOD process can come in many different forms. The main types of observations will be discussed in this section. Unknown space objects, also known as Resident Space Objects (RSOs), can be imaged using a few different methods depending on the tracking rate of the telescope. These methods will be referred to as different "modes" of observation.

2.4.1 Sidereal Rate Observations

Commercially available telescope mounts capable of observing satellites have two main options, sidereal rate, and satellite rate tracking. Sidereal rate refers to the telescope mount tracking at the observed rate of motion of the background stars. The telescope mount moves at a rate to counter the spin of the earth, keeping stars centered in the image, while other objects such as satellites that move at rates different than the background stars form lines, or streaks in the image. The main advantage of this method stems from the readily visible star fields in images for image registration (see section 3.6). Figure 2.3 shows an example image obtained using the sidereal tracking mode. Notice how the stars in the field remain point objects while there is a streak in the center of the image from a satellite passing through the image.

2.4.2 Satellite Rate Observations

An alternative method for tracking satellites is to track them at the speed at which they move through the sky. This approach requires an accurate star alignment, fast tracking mount, and applicable software. Satellite rate tracking (RSO tracking mode) can be performed either in an open loop mode or a closed loop mode. Open loop mode uses

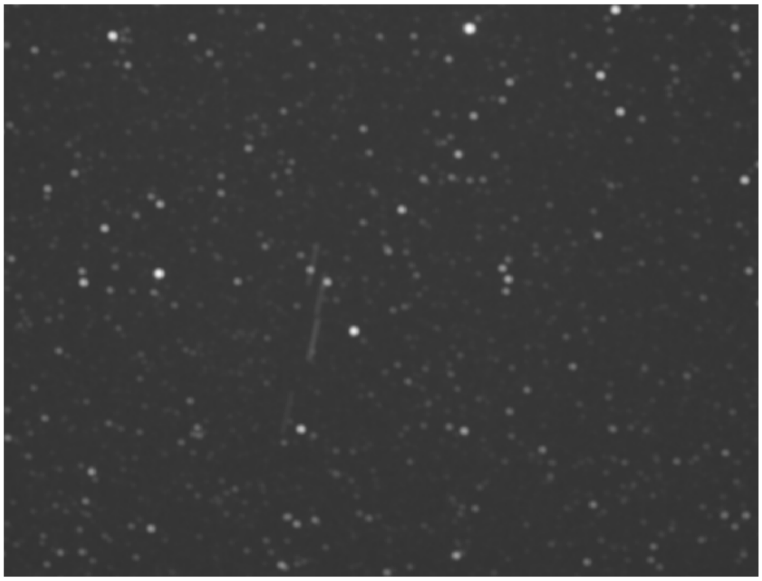


Figure 2.3. Sidereal Rate Tracking, Author's Image 2019

the known position of the observer, an accurate star alignment and an up to date state representation of the desired satellite to track. A closed loop tracking mode would use an initial satellite position to slew close before locking on with a camera to continue the tracking process. Note that any method of tracking a satellite is an *a priori* method, using the known position of a satellite to aim the telescope. We can also attempt to observe satellites and catch them through serendipitous or "lucky-imaging." Observing an area known to contain satellites such as the GEO belt is a good way to gain observations of this kind.

The advantage of RSO rate tracking is that it allows the camera sensor to gather more light of the unknown object by keeping the RSO static in the image frame. As we increase the exposure time then, we increase the signal to noise ratio of the RSO object(s). However, when RSOs streak through the image as in the sidereal tracking mode, the light from the RSO falls across many pixels and keeps moving, preventing the same increase in signal from increasing exposure time.

Figure 2.4. RSO Tracking Mode Example

2.4.3 Earth-Rate Observations

One method for observing Geosynchronous satellites is to use the earth's rotation rate as the tracking rate. Since GEO satellites orbit at the rate of the earth's motion, they appear nearly stationary in the sky. Pointing a telescope from the ground to the point in space where a GEO satellite resides results in images where the satellite remains a point source, and all stars streak through the image.

2.4.4 Point-Source Observations

Given a short enough exposure time, the observed satellite remains a point source along with the stars in the image. Because of relative motion effects, this type of observation is more likely to occur with a space-based observer. Finding RSOs in images of this type requires analyzing a set of images rather than extracting RSOs from a single image.

This mode has the disadvantage of being unable to discriminate between stars and RSOs in a single image.

Now that several different methods of obtaining measurements for performing angles-only orbit determination have been discussed, the general requirements of all angles-only orbit determination methods will be discussed.

2.5 AIOD Method Requirements

Although there are numerous methods of angles-only orbit determination, a general set of requirements for the angles-only orbit determination process can be developed based solely on the geometry of the problem. First we will setup the general vector equation from the observation geometry.

Given the location of the observation site $\mathbf{R_i}$, and the known line of sight (LOS) vector, $\widehat{\mathbf{L}}_\mathbf{i}$, to the observed target, we can derive an expression for the overall position vector to the target, $\mathbf{r_i}$. Figure 2.5 shows the geometry of a typical AIOD scenario. Equation 2.3 gives the general equation based on the geometry of each observation. The *i*th subscript

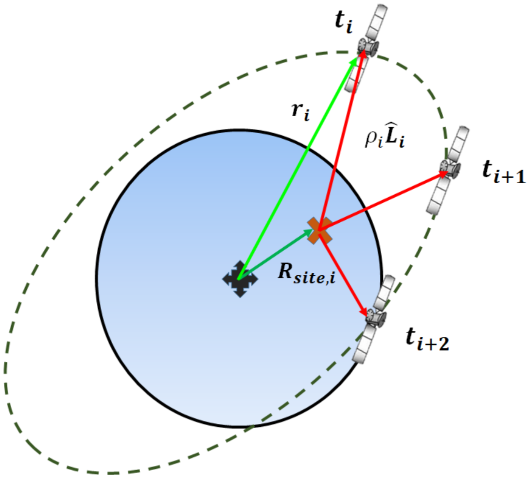


Figure 2.5. Vector Setup for AIOD Process

refers to the ith observation, for $i = 1, 2, \ldots, N$, where N is the total number of observations.

$$\mathbf{r_i} = \rho_i \widehat{\mathbf{L_i}} + \mathbf{R_i} \tag{2.3}$$

To describe the motion of a satellite in 3D space, we require a state vector with 6 independent quantities. These elements can take several different forms, the most common being a set of position and velocity vectors or the classical Keplerian Orbital Elements $\{a, e, i, \Omega, \omega, \theta\}$. Table 2.1 gives a short definition of each of the orbital elements and Figure 2.6 shows an illustration of the classic Orbital Elements. For a more in depth description of the Orbital Elements please see chapter 3 of Prussing and Conway (Prussing & Conway, 1993).

Table 2.1

Classical Orbital Elements Definitions

Element	**Symbol**	**Description**
Semi-major axis	a	Defines the size of the orbit.
Eccentricity	e	Defines the shape of the orbit.
Inclination	i	Angle between orbit angular momentum vector and Earth Earth's axis of rotation.
Argument of Perigee	ω	Angle defining perigee of the orbit with respect to the Earth's surface.
Right Ascension of the Ascending Node	Ω	Angle between Vernal Equinox direction and ascending node (intersection of the orbit and equatorial planes).
True Anomaly	ν, θ, f	Angle defining satellite's location in orbit with respect to perigee

Regardless of the type used, the six elements can be formed into a single state vector. The AIOD process attempts to determine a satellite's state from angular observations. A single image provides a maximum of two independent angles $\{\alpha_i, \delta_i\}$ that can be used to

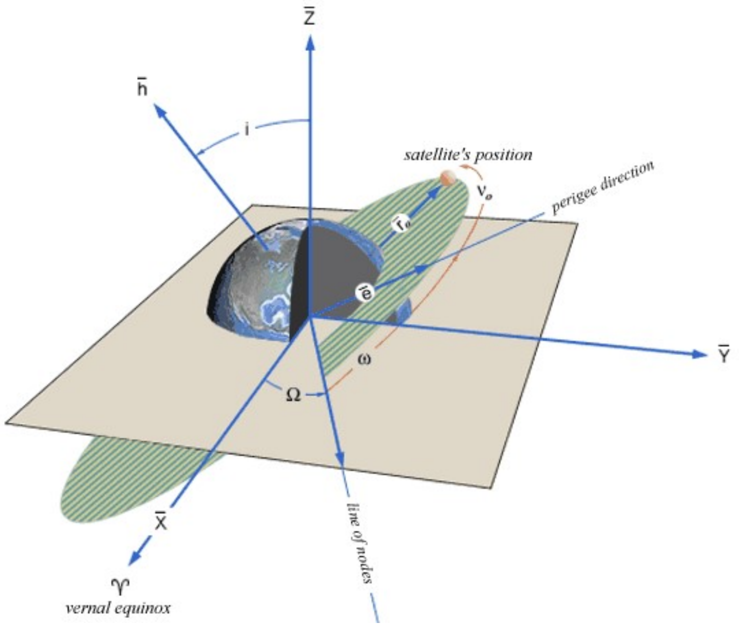


Figure 2.6. Keplerian Orbital Elements ("Orbital Elements", n.d.)

form the observed LOS vector. Uniquely determining the 6 state variables of a satellite requires a minimum of three independent optical measurements. (Three LOS measurements correspond to 6 angular measurements). Note that most historical orbit determination methods use the minimum number of observations, $N = 3$.

Given at least three observations, the task becomes that of finding the unknown range value ρ_i that corresponds to each observed LOS vector. The spacing of observations plays an important role in solution accuracy. Gauss's method (see section 2.6.2) works best with observations whose separation is $\leq 60°$. However other methods such as Escobal's (see section 2.7.1) can have observations spaced widely, even accounting for multiple revolutions of a satellite between observations.

2.5.1 Angular Data Considerations

Data for the AIOD process can be from any source providing the angular position of a satellite and the time of observation. Tycho Brahe, famous for providing the observational data Kepler used to formulate his laws of planetary motion, achieved measurement accuracy approaching one arcminute ($\frac{1}{60}$ of a degree) without the use of a telescope (Vallado & McClain, 1997). Tycho's primary instruments were the Quadrant and Sextant, non-optical devices for measuring angles. Today however, the primary source of angular data comes from images captured using optical telescopes. Sub–arcsecond ($< \frac{1}{3600}$ of a degree) angular positions can be obtained by matching the known position of stars with the stars captured in telescopic images (Lang, Hogg, Mierle, Blanton, & Roweis, 2010).

The data contained in images are captured from the electromagnetic spectrum. Cameras sensitive to visible light are the most commonly used device for observing satellites, but other wavelengths including infrared can also be used. For observing satellites, the camera attaches to the focal point of an astronomical telescope. The primary experimental setup used for this research is shown in Figure 2.7 as an illustrative example of an optical satellite observational setup.

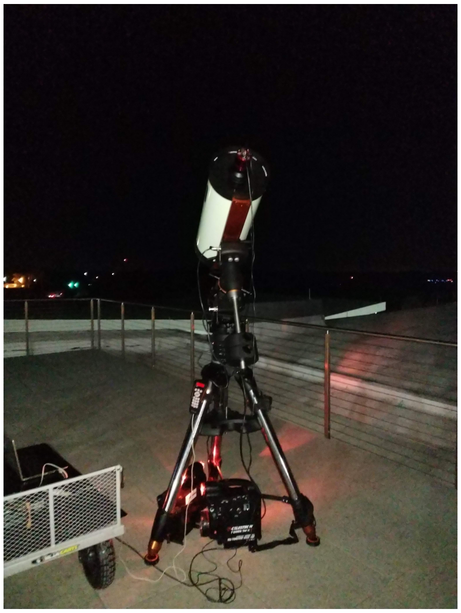


Figure 2.7. Satellite Optical Imaging Equipment Example

The quality and quantity of observational data are also important. Each image can provide at most two angular observations, leading to the aforementioned requirement of a minimum three observations. Some AIOD methods allow for the inclusion of $N > 3$ observations, which can vastly improve the orbit estimate. Utilizing $N > 3$ measurements is also necessary for going beyond an initial orbit estimate and performing precise orbit determination (POD) through differential correction or the application of sequential filtering (Schaub & Junkins, 2003). Modified versions of classical methods also exist for utilizing multiple observations to improve the initial orbit estimate. These methods are variations on classical techniques such as Gauss, Laplace, and also include a modified version of Gooding's algorithm (Henderson, Mortari, & Davis, 2010). A further discussion on quality of observation and observation noise can be found in section 2.8.

2.6 Historical Methods

The next section discusses some of the most important historical orbit determination methods. Gauss and Laplace provided two of the earliest, practical methods for solving the AIOD problem. Concerned with the motion of solar system bodies such as the planets and minor planets, these methods were not originally intended for use with the close orbiting artificial satellites of today. However, Gauss's method finds utility for satellite orbit determination given observations that are ideally spaced.

2.6.1 Laplace's AIOD Method

Pierre-Simon de Laplace (1749 - 1827), a French mathematician and astronomer, developed a deeply mathematical solution to the problem of finding an orbit from angles only. One of the first successful AIOD methods, this method works by estimating the position and velocity of the target body at the middle observation time. Laplace requires the usual minimum of three observations (two angles per observation) to independently determine the six state variables of the observed target. Escobal and Vallado both give an excellent overview of Laplace's algorithm and the summary below follows their development (Vallado & McClain, 1997; Escobal, 1976).

Laplace's Algorithm Pseudo-Code

---

1. Given values are $\widehat{\mathbf{L}}_{\mathbf{i}}, \mathbf{R}_{\mathbf{i}}, t_i$, for $i = 1,2,3$,

2. Form main vector equation from orbit/site geometry:

$$\mathbf{r}_{\mathbf{i}} = \rho_i \widehat{\mathbf{L}}_{\mathbf{i}} + \mathbf{R}_{\mathbf{i}}. \tag{2.4}$$

3. Differentiating equation 2.4 twice and rearranging terms we obtain equation 2.5. Note that we also utilize the fundamental equation of two-body motion (equation 2.2).

$$\ddot{\rho}\widehat{\mathbf{L}}_{\mathbf{i}} + 2\dot{\rho}\dot{\widehat{\mathbf{L}}}_{\mathbf{i}} + \rho\left(\ddot{\widehat{\mathbf{L}}}_{\mathbf{i}} + \frac{\mu}{r^3}\widehat{\mathbf{L}}_{\mathbf{i}}\right) = -\ddot{\mathbf{R}}_{\mathbf{i}} - \frac{\mu}{r^3}\mathbf{R}_{\mathbf{i}}. \tag{2.5}$$

4. At this point all the derivative terms are unknown, so utilizing Lagrange's interpolation formula and assuming that the middle time is zero $(t_2 = 0)$, an approximation for the derivatives of $\mathbf{r}$ and $\widehat{\mathbf{L}}_{\mathbf{2}}$ can be found (Escobal, 1976).

5. Find derivatives of the site vector (assuming same site):

$$\dot{\mathbf{R}}_{\mathbf{2}} = \omega_{\mathbf{earth}} \times \mathbf{R}_{\mathbf{2}} \tag{2.6}$$

$$\ddot{\mathbf{R}}_{\mathbf{2}} = \omega_{\mathbf{earth}} \times \dot{\mathbf{R}}_{\mathbf{2}}. \tag{2.7}$$

6. If observations come from multiple sites, use Herrick-Gibbs approach. See section 6.6.2 of (Vallado & McClain, 1997) for more details.

7. Now we have a matrix vector equation with unknowns $\rho, \dot{\rho}, \ddot{\rho}$, and r.

$$\begin{bmatrix} \widehat{\mathbf{L}}_\mathbf{i} & 2\dot{\widehat{\mathbf{L}}}_\mathbf{i} & \ddot{\widehat{\mathbf{L}}}_\mathbf{i} + \frac{\mu}{r^3}\widehat{\mathbf{L}}_\mathbf{i} \end{bmatrix} \begin{bmatrix} \ddot{\rho} \\ \dot{\rho} \\ \rho \end{bmatrix} = - \begin{bmatrix} \ddot{\mathbf{R}}_\mathbf{i} + \frac{\mu}{r^3}\mathbf{R}_\mathbf{i} \end{bmatrix} \tag{2.8}$$

8. By using determinants and applying Cramer's rule, an eighth order polynomial in the unknown vector magnitude r can be formed. Note that an initial guess must be made for r before beginning the iteration process. Prussing and Conway show that the real root gives the desired correct root to the equation, and lets us find the position vector for the middle time (Prussing & Conway, 1993).

9. The last step is to find the velocity at the middle time:

$$\mathbf{v_2} = \dot{\rho}_2\widehat{\mathbf{L}}_\mathbf{2} + \rho\dot{\widehat{\mathbf{L}}}_\mathbf{2} + \dot{\mathbf{R}}_\mathbf{2} \tag{2.9}$$

10. The full state vector of the satellite has now been found at the time of the middle observation $(\mathbf{r_2}, \mathbf{v_2})$.

While Laplace's method provides a solution for orbit determination, its use is found primarily for distant planetary, or comet observations. The accuracy of the standard method for earth orbiting satellites is unreliable. To overcome this, more observations and higher order terms for the Lagrange interpolation formulae must be used (Escobal, 1976). Recent modifications to Laplace's original algorithm have overcome these difficulties by using extra observations and applying differential correction techniques (Branham, 2005). However, Gooding's method (discussed in section 2.7.2) remains the most popular method in use today (Karimi & Mortari, 2014).

2.6.2 Gaussian Determination

Charles Frederic Gauss (1777-1855) provided the first accurate, reliable, yet computationally feasible method for determining orbits from angles only. Motivated by the discovery of the dwarf planet Ceres on January 1, 1801, and its subsequent disappearance behind the sun a little over a month later, Gauss had a very short arc of observational data to work with. However, he confidently asserted that Ceres would reappear after it had gone around the sun late in 1801 or early in 1802. Later that year on December 7, 1801, Ceres was observed very near Gauss's predicted location, thus sealing his fame as an astronomer.

For applications to earth orbiting satellites, Gauss's method provides surprisingly good results for closely spaced observations (observations $< 60^\circ$ apart). Vallado remarks that the best results can be achieved for this method when using data spread no more than 10° apart (Vallado & McClain, 1997). The heart of Gauss's method relies on an accurate approximation of the Lagrange–Gibbs f and g series as well as the assumption that the observed line of sights $\widehat{\mathbf{L_i}}$ are coplanar.

Lagrange–Gibbs f and g Solutions

Before continuing to Gauss's orbit determination algorithm, some background material on the Lagrange–Gibbs f and g functions must be presented. An orbit can be described by the set $\{\mathbf{r},\mathbf{v}\}$ at any arbitrary time t. A reasonable conclusion from this fact is that an orbit can be described at any time by a linear combination of $\{\mathbf{r},\mathbf{v}\}$ at some specified epoch time t_0. The Lagrange–Gibbs, f and g functions are scalar valued functions that allow the position and velocity of a satellite at any time to be written as a linear combination of the position and velocity at a specified time. Note that the functions are approximations of an infinite series definition.

The f and g solution begins with the assumption that orbital motion is purely planar motion. From this assumption, we can write equations 2.10 and 2.11, which give the position and velocity at any time in an orbit as a linear combination of the position and velocity at some initial time t_0 using the scalar valued functions f and g and their derivatives.

$$\mathbf{r}(t) = f\mathbf{r}(t_0) + g\mathbf{v}(t_0) \tag{2.10}$$

$$\mathbf{v}(t) = \dot{f}\mathbf{r}(t_0) + \dot{g}\mathbf{v}(t_0) \tag{2.11}$$

The f and g functions are scalar valued functions that depend on the geometry of the orbit problem. In order to use these functions, one must be able to calculate the orbital quantities used in the definitions, such as the semi-major axis (a), eccentric anomaly (E), and position vector magnitude (r). Note that $\widehat{E} = E - E_0$ is defined as the change in eccentric anomaly between the epoch times t and t_0. For a more thorough treatment of the f and g solutions please refer to Battin sections 3.4 and 3.6 (Battin, 1999).

$$f = 1 - \frac{a}{r_0}\left(1 - cos\left(\widehat{E}\right)\right) \tag{2.12}$$

$$\dot{f} = \frac{\sqrt{\mu a}}{r r_0} sin\left(\widehat{E}\right) \tag{2.13}$$

$$g = (t - t_0) + \frac{a^{3/2}}{\mu}\left(sin\left(\widehat{E}\right) - \widehat{E}\right) \tag{2.14}$$

$$\dot{g} = 1 - \frac{a}{r}\left(1 - cos\left(\widehat{E}\right)\right) \tag{2.15}$$

With the f and g functions defined, we now have all the necessary background to examine Gauss's orbit determination technique. The following algorithmic summary highlights the key portions of Gauss's method.

Gaussian Determination Pseudo-Code

---

1. Given values are $\widehat{\mathbf{L_i}}$, $\mathbf{R_i}$, t_i, for $i = 1,2,3,$.
2. Assume that all position vectors are coplanar, then:

$$c_1\mathbf{r_1} + c_2\mathbf{r_2} + c_3\mathbf{r_3} = \mathbf{0}. \tag{2.16}$$

3. Define expressions for c_1, c_2, c_3, with $c_2 \neq 0$, by taking cross products of equation 2.16.
4. Utilize f and g functions to write the position vectors in terms of time t_2. (The f and g Lagrange–Gibbs equations allow any satellite position vector to be written as a linear combination of the position and velocity at any chosen time (Schaub & Junkins, 2003)).

$$\mathbf{r_i} = f_i\mathbf{r_2} + g_i\mathbf{v_2}, i = 1,3 \tag{2.17}$$

5. Rescale the coefficients using equation 2.17, while letting $c_2 = -1$.

$$c_1 = \frac{g_3}{f_1 g_3 - f_3 g_1} \tag{2.18}$$

$$c_3 = \frac{-g_1}{f_1 g_3 - f_3 g_1} \tag{2.19}$$

6. Substitute in the series form of the f and g functions to write c_1, c_2, c_3 again.
7. From the observation geometry, equation 2.3, substitute into equation 2.16, and rearrange:

$$c_1\mathbf{r_1} + c_2\mathbf{r_2} + c_3\mathbf{r_3} = -c_1\mathbf{R_1} - c_2\mathbf{R_2} - c_3\mathbf{R_3}. \tag{2.20}$$

8. At this point we can include the observed LOS vectors, $\widehat{\mathbf{L_i}}$, to form a matrix vector equation analogous to that used in Laplace's method (equation 2.8).

$$\begin{bmatrix} \widehat{\mathbf{L_1}} & \widehat{\mathbf{L_2}} & \widehat{\mathbf{L_3}} \end{bmatrix} \begin{bmatrix} c_1\rho_1 \\ c_2\rho_2 \\ c_3\rho_3 \end{bmatrix} = \begin{bmatrix} \mathbf{R_1} & \mathbf{R_2} & \mathbf{R_3} \end{bmatrix} \begin{bmatrix} -c_1 \\ -c_2 \\ -c_3 \end{bmatrix} \tag{2.21}$$

9. Now we can use matrix solution methods to solve the equations simultaneously. The resulting 8^{th} order polynomial is similar to that obtained during the solution to Laplace's method with the added difficulty that there are several real, positive roots that must be discriminated between to find the correct solution.

10. Iteration: Go back to step 5, this time using the actual f and g functions instead of the series approximations. Solve the matrix relation in equation 2.21 again and repeat until the slant-range values converge (Vallado & McClain, 1997).

Overall Gauss's method provides an accurate and robust method for satellite orbit determination given ideally spaced measurements. There exist extensions to Gauss's method allowing the use of multiple observations.

2.6.3 Lambert's Problem

Though not strictly an angles-only orbit determination process, Lambert's Problem is essential to many OD processes, including that of Gooding discussed in section 2.7.2. Finding the orbit between two positions given the time of flight between the two positions is known as Lambert's Problem. Though the first numerical solution was published a year after Lambert’s death by Joseph Louis Lagrange, the solution to this two-point boundary problem still bears Lambert’s name.

Johann Heinrich Lambert (1728-1777), a Swiss born mathematician made important contributions to many areas in mathematics. Especially fond of geometry, he studied map projections, hyperbolic functions, and conic sections. Utilizing his fantastic geometrical reasoning, Lambert proposed the following theorem stating that the time of flight between two points on an orbit depends purely on the geometry involved.

Lambert’s Theorem

> The orbital transfer time depends only upon the semimajor axis, the sum of the distances of the initial and final points of the arc from the center of force, and the length of the chord joining these points (Vallado & McClain, 1997).

The geometry of the two-point boundary value problem (BVP) can be seen in Figure 2.8 (Schaub & Junkins, 2003). The crux of finding a solution to Lambert’s problem is solving for the orbital parameters that connect the two points on the orbit in the specified transfer time. Lambert’s problem also has applications in orbital maneuvering and finding the transfer orbit between two desired orbits.

There are many excellent algorithms for solving Lambert’s problem. However, there are key pieces applicable to any Lambert solver. The different solutions each have their own merit. Battin presents an elegant universal variables solution while Gooding provides a more modern numerical approach (Battin, 1999; Gooding, 1990). Dario Izzo revisited Lambert’s problem recently and provides an implementation with the accuracy of Gooding’s but is less computationally demanding (Izzo, 2015). (Note that Izzo’s approach to

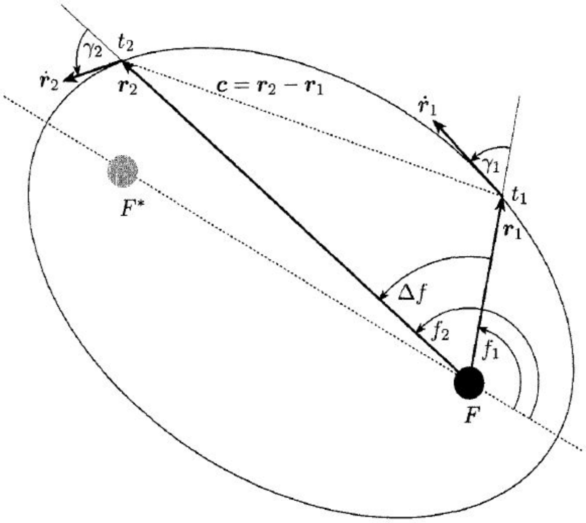


Figure 2.8. Geometry of Lambert's Problem (Schaub & Junkins, 2003)

solving Lambert's Problem is used in the implementation of Gooding's AIOD method utilized by this research.)

Lambert's original approach to the problem focuses on finding the minimum energy orbit using only the geometry of ellipses (Vallado & McClain, 1997). The core of every Lambert solver is the numerical method employed to solve the general time of flight equation, which is referred to by some as *Lambert's Equation* (equation 2.22).

$$t = \sqrt{\frac{a^3}{\mu}}(\alpha - \sin\alpha - (\beta - \sin\beta)) \tag{2.22}$$

Note that the parameters α and β in equation 2.22 depend only on the ellipse geometry. From equation 2.22 we can see that Lambert's theorem holds true, as the time of flight depends solely on the geometric quantities of the elliptical orbit. The pseudo-algorithm below gives a general approach to solving Lambert's problem. Given two position vectors and the their times, the goal is to find the velocities at those points. Prussing and

Conway's generalized approach provides the easiest to understand derivation and will be followed in the psuedo-code below (Prussing & Conway, 1993).

Lambert's Problem Pseudo-Code

---

1. Given values are: $\mathbf{r_1}, \mathbf{r_2}, t_1, t_2$, with $t_2 > t_1$

2. Define several quantities needed for later calculations.

   Chord between position vectors:

   $$c = \frac{\mathbf{r_2} - \mathbf{r_1}}{\|\mathbf{r_2} - \mathbf{r_1}\|}. \tag{2.23}$$

   Transfer angle between $\mathbf{r_1}$ and $\mathbf{r_2}$:

   $$\theta = f_2 - f_1. \tag{2.24}$$

   Semiperimeter (s) of the triangle defined by the focus and the two position vectors:

   $$s = \frac{r_1 + r_2 + c}{2}. \tag{2.25}$$

   Parameters α and β:

   $$\sin\left(\frac{\alpha}{2}\right) = \sqrt{\frac{s}{2a}} \tag{2.26}$$

   $$\sin\left(\frac{\beta}{2}\right) = \sqrt{\frac{s-c}{2a}}. \tag{2.27}$$

3. Determine if the transfer orbit is elliptic or parabolic by finding the parabolic transfer time t_p:

   $$t_p = \frac{\sqrt{2}}{3\sqrt{\mu}}\left[s^{\frac{3}{2}} - sgn(\sin\theta)(s-c)^{\frac{3}{2}}\right]. \tag{2.28}$$

4. If $t_2 - t_1 > t_p$, the transfer orbit is elliptic; otherwise, the orbit must be parabolic or hyperbolic
5. Calculate the minimum energy transfer time, t_m, which allows the correct value of α to be found

$$t_m = \sqrt{\frac{s^3}{8\mu}}\left(\pi - \beta_m + \sin\beta_m\right). \tag{2.29}$$

6. Find correct values for the parameters α and β, using the value of θ.
7. Solve Lambert's Equation (equation 2.30) using an iterative numerical approach:

$$t = \sqrt{\frac{a^3}{\mu}}(\alpha - \sin\alpha - (\beta - \sin\beta)). \tag{2.30}$$

   Gooding and Battin both present excellent numerical methods for finding the iterative solution for the semimajor axis a (Battin, 1999; Gooding, 1990). A choice must be made on whether universal application or computational ease is the desired goal when choosing a Lambert solver.

8. Use the unique solution for a to find the velocities at t_1 and t_1
9. The orbit is now fully defined by either $\{\mathbf{r_1}, \mathbf{v_1}\}$ or $\{\mathbf{r_2}, \mathbf{v_2}\}$

Solving Lambert's Problem is an integral part of Gooding's AIOD method presented in section 2.7.2. Essentially Gooding uses Lambert to iterate on different guesses for unknown measurement ranges.

2.7 Recent AIOD Methods

This section reviews some of the more recent algorithms developed for angles-only orbit determination. This includes a summary of Escobal's Double-r iteration, Gooding's method, and a modified version of Gooding's method extended to use multiple measurements. Unlike the methods previously discussed, these modern methods were derived with their application to finding the orbit artificial satellites in mind as opposed to historical methods whose main goal was finding the orbit of comets and minor planets.

Recently there has been much research into finding the orbit of satellites by Angles-Only techniques. The passive nature of optical observations makes AIOD an ideal choice for SSA applications thanks to the low power, low cost systems available. Techniques varying from particle filters to genetic algorithms have been applied to the solving the AIOD problem. New methods involving the f and g series coefficients have also been developed (Karimi & Mortari, 2014). The techniques of Escobal and Gooding have been chosen to be examined in more detail.

2.7.1 Escobal Double r-Iteration

Escobal uses a combination of mathematical and dynamical techniques to arrive at a solution to the AIOD problem. Escobal's method has an important advantage over Gauss's in that observations are not confined to small angular separations. The main contribution of double-r comes from allowing for solution to the multi-revolution problem. There are four main steps to the algorithm. First a bound is put on the initial guess for the solution. The second step bounds the intermediate guesses. The third step then begins the actual iterative process, which attempts to align the transfer times with the observed line of sight directions. Finally a differential correction scheme is applied to find the final values for the middle position $(\mathbf{r_2})$ and velocity $(\mathbf{v_2})$ vectors.

Escobal Double r-Iteration Pseudo-Code

---

1. Given values are $\widehat{\mathbf{L_i}}, \mathbf{R_i}, t_i$, for $i = 1,2,3,$.
2. Find $\mathbf{r_2}$ and $\mathbf{v_2}$.
3. Guess values for the position magnitudes r_i at times t_1 and t_2, to form an initial guess for the LOS ranges $\rho_i, i = 1,2,3$.
4. From the assumed values of r_1, r_2, calculate $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}$.
5. Calculate the time interval between observations, use the difference in true anomalies and Kepler's equation (2.31) to form an expression for the time intervals as a function of r_1 and r_2:

$$M = E - e \sin E. \tag{2.31}$$

$$\bar{t}_3 - \bar{t}_2 = \frac{M_3 - M_2}{n} \tag{2.32}$$

$$\bar{t}_1 - \bar{t}_2 = \frac{M_1 - M_2}{n}. \tag{2.33}$$

6. Define functions F_1 and F_2 that must be forced to zero to match the assumed modified times: $(\tau_1 = k(t_1 - t_2), \tau_2 = k(t_3 - t_2))$.

$$F_1 \equiv \tau_1 - k\frac{M_1 - M_2}{n} \tag{2.34}$$

$$F_2 \equiv \tau_2 - k\frac{M_3 - M_2}{n} \tag{2.35}$$

7. Apply numerical correction to the assumed magnitudes of the radius vectors. Applying a two-dimensional Taylor series expansion and varying during each iteration successively r_1, then r_2 (hence the name double r-iteration), lets every third pass through the algorithm provide an improvement of the estimated value for r_i (Escobal, 1976).

8. When a specified convergence tolerance for the value of r_i is reached, we can form the f and g coefficients to directly find $\dot{\mathbf{r}}_\mathbf{2}$.

$$\dot{\mathbf{r}}_\mathbf{2} = \frac{\mathbf{r_3} - f\mathbf{r_2}}{g} \tag{2.36}$$

9. The orbit is now fully defined by the fundamental set r_2 and $\dot{\mathbf{r}}_\mathbf{2}$.

2.7.2 Gooding's Method

R. H. Gooding published an effective and robust new iterative method for orbit determination from a minimum of three angles-only observations as a technical report in 1993 (Gooding, 1993). Published again as a journal article in 1996, today Gooding's method remains one of the most widely used AIOD methods, owing to both its relative computational ease, and robustness for most orbit geometries (Gooding, 1996; Karimi & Mortari, 2014). Gooding's method is the primary AIOD method used for this research, and a more thorough algorithmic description and derivation of this method is presented than for the other AIOD methods described above.

Vallado offers a good overall evaluation of Gooding's method applied to the problem of space-based observations (Vallado, 2010). Others have used Gooding for ground based observation studies of GEO satellites through STK's Orbit Determination Tool Kit (ODTK), which uses an implementation of Gooding's method (Montojo, López Moratalla, & Abad, 2011). Gooding's method has been extensively used for nearly all orbit regimes (LEO, MEO, and GEO), and has been shown to provide an accurate and robust initial orbit estimate for most scenarios (Martínez, Águeda Mat, Grau, Fernndez Snchez, & Aivar Garca, 2012).

Schaeperkoetter performed an exhaustive comparison of classical AIOD methods and concluded that Gooding consistently provided the best orbit estimate for nearly every observational scenario (Schaeperkoetter, 2011). Schaeperkoetter does note that the one exception is for polar orbits, where the solution accuracy greatly depends on the observational time intervals (Schaeperkoetter, 2011). For the case of a space-based observer, Gooding provides accurate results if an accurate initial estimate of the range can be provided (Schaeperkoetter, 2011). Given that the state of an observer satellite is usually known to acceptable accuracy, suitable bounds can be placed on the initial range estimate and thus Gooding can provide a good orbit estimate for space-based observations.

Gooding's Assumptions

Gooding's only assumption is that the target satellite is moving under unperturbed Keplerian motion. No other assumptions are made regarding the observations or motion of the satellite. In fact one of Gooding's original intentions was to develop a method that was as independent as possible from particular details of the observer (Gooding, 1993). As such, his method holds the advantage of being able to utilize measurements with any spacing between them, from multiple sites, including measurements from multiple revolutions of the satellite. Gooding's method also does not exhibit the characteristic singularities that Gauss's and Laplace's methods suffer from for co-planar and circular orbits (Henderson et al., 2010).

Gooding Method Overview

For the following development, Gooding's original notation will be followed (Gooding, 1993). Gooding begins by setting up the vector geometry of the AIOD problem using the observed line of sight vector($\lambda_\mathbf{j}$), observer position ($\mathbf{R_j}$), and satellite position ($\mathbf{r_j}$), at three distinct times, t_j $(j = 1,2,3)$. Assuming the unknown range to be ρ_j, then, the observation geometry at t_j is given by equation 2.38.

$$\rho_j \lambda_\mathbf{j} = \boldsymbol{\rho}_\mathbf{j} \tag{2.37}$$

$$\boldsymbol{\rho}_\mathbf{j} = \mathbf{r_j} - \mathbf{R_j} \tag{2.38}$$

An initial guess for the edge ranges must be supplied, ρ_j, for $j = 1,3$, as well as a guess on whether the orbit is prograde or retrograde. Using the initial range estimates, Gooding solves Lambert's Problem (detailed above in Section 2.6.3) to arrive at an estimated position vector at t_2. It should be noted that Gooding utilized his own universal Lambert orbital boundary value problem solver, but any suitable Lambert Solver can be substituted by the user (Gooding, 1990). Given this modularity of the method, Dario Izzo's approach to solving Lambert's problem is utilized in this research (Izzo, 2015).

If the position value is correct for the orbit, then the unit vector $\mathbf{\hat{r}_2}$ would coincide perfectly with the second measured LOS, $\lambda_\mathbf{2}$. However, with measurement errors and an imperfect guess for the ranges, this is usually not the case and an iterative process must be followed. Gooding chose to use a Newton-Raphson approach to iteratively correct the range values until the solution converges. Overall Gooding's method turns out to be rather robust to different range guesses, deals with multiple revolutions, and remains computationally light (Gooding, 1996). Following is a more detailed algorithmic description of Gooding's AIOD method.

Gooding Algorithm Psuedo-Code

```
1. Given values are $\lambda_\mathbf{j}, \mathbf{R_j}, t_j$, for $j = 1,2,3$.
```

2. Utilize measured LOS vectors, $\hat{\boldsymbol{\lambda}}_{\mathbf{j}}$ and given information, form the main vector equation of the AIOD problem.

$$\rho_j \hat{\boldsymbol{\lambda}}_{\mathbf{j}} = \boldsymbol{\rho}_{\mathbf{j}} = \mathbf{r_j} - \mathbf{R_j} \tag{2.39}$$

3. Gooding uses the notation that $t_{jk} = t_k - t_j$, and assumes that the times take the form t_j such that $t_1 < t_2 < t_3$. By defining t_{jk}, and given only a set of three measurements this gives a "special role" for the measurement at t_j for $j = 2$, which Gooding chooses as the time to compare the iterative results with.
4. Guess a value for the edge ranges, ρ_1, and ρ_3. At this point any knowledge of the observed satellites orbit regime can be used to bound the guess on the range values to a suitable number. If no information is available, assume a reasonable guess as a multiple of earth's radius.
5. Assume a direction for the orbit being estimated (prograde or retrograde)
6. Begin iterative loop: until tolerance for range changing between iterations is reached.

   (1) Use the assumed ranges at t_1 and t_3 to form the satellite position vectors r_1, r_3 from equation 2.39

   (2) Solve Lambert's problem with r_1, r_3, t_{31} to provide an estimate for r_2

   (3) Define target functions f and g for use in an iterative correction process. Gooding uses the line of sight at t_2, $\hat{\boldsymbol{\lambda}}_{\mathbf{2}}$, to identify the normal to a family of parallel oriented planes. The intersection of $\hat{\boldsymbol{\lambda}}_{\mathbf{2}}$ with each plane is chosen as the origin of a unique coordinate system within that plane.

   (4) Compute the origin (P_c) of the aforementioned plane. Then the target functions are simply the coordinates (f, g) in the plane. Note that for each iteration

the axis system is redefined, so that for different values of the edge ranges ρ_1, ρ_3, different values of the target functions result. (Note Gooding simplifies notation with $\rho_1 = x, \rho_2 = y$ for the following functional analysis).

(5) At this point Gooding chose to use a Newton-Raphson procedure to correct the assumed values for ρ_1 and ρ_3.

$$\begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = - \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}^{-1} \begin{pmatrix} f \\ g \end{pmatrix} \tag{2.40}$$

(6) Gooding defines $\delta x, \delta y$ as the corrections for the estimate of a pair of roots of the two equations:

$$f(x,y) = 0 \tag{2.41}$$

$$g(x,y) = 0 \tag{2.42}$$

(7) Gooding assumes that $g = 0$. This allows us to break equation 2.40 into two simpler equations, where, $D = f_x g_y - f_y g_x$, is the determinant of the derivative matrix in equation 2.40.

$$\delta x = -D^{-1} f g_y \tag{2.43}$$

$$\delta y = -D^{-1} f g_x \tag{2.44}$$

(8) Gooding finds the partial derivatives of f and g by truncating a Taylor series expansion of f and g after the linear terms. The truncation introduces some error into the process, but as long as the determinant matrix is well conditioned, and provided the initial values for x and y are not too far from the solution, the iteration process displays quadratic convergence (Gooding, 1993). If the starting point is too far off the solution, or the derivative matrix is ill conditioned,

then a large number of iterations may be required for convergence, or the solution may not converge at all (Henderson et al., 2010). Gooding in his original paper, proposed using the second derivatives and a modified Newton-Raphson, or Halley, process to give the iteration process a cubic convergence rate rather than the quadratic rate of using only the first derivatives.

(9) The correction procedure used by Gooding sometimes provides multiple solutions, and Gooding introduces another Newton like procedure that “steers” the convergence away from already known solutions so as to obtain all the available solutions. Gooding also handles the multi-revolution problem by changing increment sizes upon a failure for solution to Lambert’s problem after a large Newton-correction to the ranges.

7. Once the range values have reached a specified tolerance, the process is complete and the orbit is known by the set $\{\mathbf{r_2}, \mathbf{v_2}\}$ resulting from the final Lambert solution.

2.7.3 N-Gooding

While providing excellent results given ideally spaced measurements, Gooding’s original method is limited to the use of only three measurements. When presented with a dataset having $N > 3$ measurements, some method would need to be used to select the optimal set of three measurements for the AIOD process. However, this is a non-trivial process given a large data set. Investigating all combinations of measurements is one method, but is computationally very expensive. Instead of choosing an optimal set of three measurements, Henderson et. al. propose a version of Gooding’s method extended to utilize $N > 3$ measurements (Henderson et al., 2010). This approach uses a different cost function minimization than Gooding's original algorithm, seeking to minimize the residuals of a cost function based on the angular difference between the observed and computed line of sights. A genetic algorithm was utilized to minimize the cost function. It is worth noting that any suitable cost function designed by the user can be substituted for minimization. As N-Gooding follows most of the same steps as the original Gooding algorithm, a short pseudo-code is presented below rather than a full derivation of Gooding again.

N-Gooding Pseudo-Code

1. Given values are $\boldsymbol{\lambda}_{\mathbf{j}}, \mathbf{R}_{\mathbf{j}}, t_j$, for $j = 1,2,3,.$
2. Assume a value for the edge ranges, ρ_1 and ρ_N
3. Iterate until maximum iterations or a tolerance are reached:
4. Generate an estimated orbit by solving Lambert's problem using the edge positions, r_1, r_N, and t_{N1}
5. Compute measurement error in the position measurement of the spacecraft. Measurement error is taken to be the cross product between the computed and measured LOS unit vectors.

   $$error_j = \hat{\lambda}_j \times \tilde{\lambda}_j \tag{2.45}$$

6. For small angles, the *sine* function approximates the angle itself. Therefore the cost function to be minimized becomes the sum squared of the angles (ϕ_j) between the LOS unit vectors:

   $$J = \sum_{j=1}^{N} \phi_j^2 \tag{2.46}$$

7. The N-Gooding algorithm is modular however, and any cost function of the form shown in equation 2.47 can be minimized.

$$J = f\left(\sum_{i=1}^{N} w_i g_i\left(\hat{\rho}_i, \tilde{\rho}_i\right)\right) \tag{2.47}$$

8. Iterate the values for the edge ranges ρ_1 and ρ_N.
9. Once tolerance or maximum iterations are reached, the orbit has been determined and can be represented by the set $(\mathbf{r_1}, \mathbf{v_1})$.

The ability to incorporate $N > 3$ measurements gives the N-Gooding method the capability of potentially providing a more accurate initial orbit estimate. Given that an AIOD solution is traditionally the starting place for differential correction, batch filtering, or other methods of precise orbit determination, it is desirable to arrive at the most accurate first estimate as possible. As mentioned by Henderson, the choice of an error function in the current implementation of N-Gooding was somewhat arbitrary, and different error measurements could lead to improved performance of the algorithm (Henderson et al., 2010).

2.8 Noise

Once an orbit estimate has been found, now the question becomes, “how good is it?.” In order to answer the question as to the quality of an orbit solution, we must know something about any noise in the measurements used in the OD process. Given knowledge of noise, then we can develop an estimate of the orbit estimate’s covariance. Classical orbit determination methods with 3 measurements are deterministic processes assuming zero measurement noise. Given a set of observations, and a proper guess to initialize the routine, the same solution will be reached if we run the process with the same inputs again. However, including multiple observations beyond the initial three measurements changes the orbit determination process deterministic solution into a probabilistic distribution. Even in the three measurement case, the result is not exact, but rather represents an estimate with some unknown amount of covariance. Therefore, covariance information must be computed separately and propagated to the measurement times in order to assign a level of confidence to the solution.

Whenever we perform orbit determination and wish to include covariance information, we must know the possible sources of noise in the process. For the target system of this research (small aperture optical systems) noise comes from a variety of physical, electrical, and digital sources. This type of system differs from typical government run space surveillance systems. While government systems such as AMOS have large budgets and expensive permanent facilities, small mobile systems face a variety of challenges not seen by these larger facilities.

2.8.1 Physical Noise Sources

First, noise sources from a physical sense must be considered. With the system used in this research (see section 3.11), which is not permanently mounted, variations in setup location affect the observations from session to session. Taking measures such as marking setup location can help alleviate some of these difficulties but do not remove them. Less than ideal conditions from the observation site also affect images. Light pollution plays a significant factor in diminishing the quality of captured images. Weather, clouds, and rain limit observation times. While some level of cloud cover may be acceptable for gathering data, any clouds obscuring the view of the telescope will block some of the light from the desired target. Another physical factor owing to the type of setup used is physical vibrations. Vibrations from walking around near the telescope, or from a nearby road can shake the telescope and ruin a given image. Wind also can induce vibrations that will cause image jitter and degrade the resulting image.

Atmospheric Effects

Ground-based observations face the significant challenge of always observing through Earth's atmosphere. Given that the targets of interest are satellites, which can be quite dim, this at the very least limits observations to during the night. Also, because optical observations are simply pictures of the night sky, then another prerequisite for being able to gather data is clear weather. The atmosphere is always in motion. Both high level and ground level winds can cause problems. Higher atmospheric winds accentuate the scattering of starlight passing through the turbulent air, while strong ground level winds can introduce unwanted vibration of the telescope resulting in blurred images.

Optical Noise

Optical noise sources must also be considered. Optical defects can distort the resulting image. Optical defects in the telescope lenses or mirrors lead to imperfectly focused images. The glass used to manufacture optical elements have differing qualities (and increasing prices with better qualities such as lower dispersion). Other problems such as dust or dirt on the optical elements result in image artifacts, degrading the image quality. Achieving perfect focus of all wavelengths of visible light might not be possible given the different optical properties of the glass used. The physical movement of a telescopes mirror or lens assembly to achieve focus is often done through a geared bearing mechanism that often does not allow the reaching of perfect focus.

2.8.2 Thermal Noise

Cameras use electronic sensors, which produce some amount of heat when operating. Thermal signal can lead to thermal noise introducing extraneous signals into the resulting image. Also, the temperature of the camera sensor greatly affects the noise readout level of a given camera. To combat this kind of noise, a cooled camera can be used. Cooled cameras are capable of chilling the sensor to a desired setpoint that minimizes thermal noise. Figure 2.9 shows the camera used for this research. The ASI1600M camera is a 16 megapixel mono CMOS Thermo-electrically cooled camera that is commercially available.

Figure 2.9. ASI1600M Cooled CMOS Camera (ZWO Company, 2016)

2.8.3 Electronic Noise

Electronic noise is noise added in the process of capturing, reading, and saving an image from the camera. Read noise and dark current are the main forms of electronic noise inherent to all digital cameras. Read noise is the noise added in the process of converting the signal registered in a camera's sensor into an actual pixel intensity (On, 2016). Dark noise is the result of electrons generated at the sensor measurement (pixel) sites by thermal variations of the sensor (Teledyne Photometrics, n.d.). Since these types of noise are inherent to the sensor, measures can be taken in the camera design to minimize the read noise and dark current of the camera.

2.8.4 Digital Noise

One more source of noise that must be considered is any noise resulting from the digital processing of images. Computers are limited in the precision they can use to represent images. Also, all the processing applied to an image is limited in the precision of calculations. In essence, there will always be some amount of round-off error present. For example, the calculations used to find the locations of RSOs in an image use double precision (16 bit) in MATLAB, and are only accurate to 16 decimal places. This might not seem like much, but when we consider the compounding effect of small errors in the localization of an RSO in an image that add up to a few pixels, this translates to much larger angular errors of the actual observation from the image.

2.8.5 Covariance

Once a number value can be estimated for the total noise in the measurements, an initial estimate of the measurement covariance can be developed. The covariance matrix gives information about the joint and independent variation between two or more variables. Considering covariance for a 2D case, taking x_e, y_e to be two estimated states, then the covariance matrix can be calculated by using equation 2.48, where $\sigma_{x_e}, \sigma_{y_e}$ are the $1-\sigma$ variances of the estimated states and $\rho_{x_e y_e}$ is the cross correlation coefficient.

$$P = \begin{bmatrix} \sigma_{x_e}^2 & \rho_{x_e y_e}\sigma_{x_e}\sigma_{y_e} \\ \rho_{x_e y_e}\sigma_{x_e}\sigma_{y_e} & \sigma_{y_e}^2 \end{bmatrix} \tag{2.48}$$

In the context of orbit determination, the covariance matrix relates how well the final orbit solution can be trusted. Given that the final AIOD solution contains 6 states, the covariance matrix $P(\mathbf{x_e})$ will be a 6×6 matrix. The final orbit solution will then have a an uncertainty distribution that is 6-dimensional. In general an uncertainty distribution is referred to as an error ellipsoid. Common error ellipsoids for the orbit determination process can be formed by considering the position and velocity states separately and forming two separate, 3-dimensional error ellipsoids. The elements of $P(\mathbf{x_e})$ can be used to give the size and orientation of the error ellipsoid. An example error distribution for the initial position vector found by running a Monte-Carlo simulation of an AIOD solution is shown in Figure 2.10.

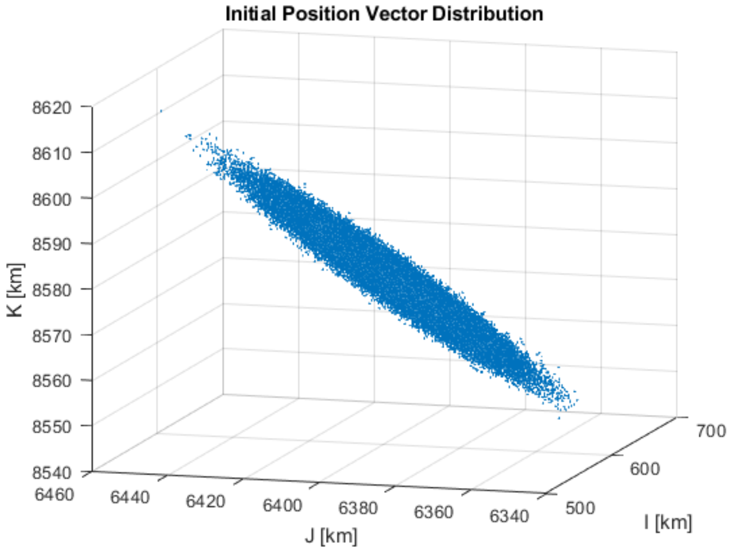


Figure 2.10. Position Vector Error Ellipsoid

Obtaining the error ellipsoid can be done through two primary means. The error ellipsoid can be found by running Monte-Carlo trials to generate the error distribution or by using a transformation method such as the Unscented Transform. Running large numbers of trials is generally computationally expensive, making the choice of Monte-

Carlo trials unlikely. Another method involves taking the known error characteristics before running the AIOD process and using that information to compute the covariance for the AIOD solution. This can be done through the use of the Unscented Transform which is discussed in the next section.

2.8.6 Unscented Transform

The Unscented Transform (UT) provides a means of transforming the mean and covariance from one frame to another through any arbitrary non-linear transformation process (Julier, Uhlmann, & F. Durrant-Whyte, 2000). The UT uses a set of statistically significant points to approximate the mean and covariance. This set of "sigma points", are then fed through a non-linear transformation, the orbit determination process for example, before backing out a new mean and covariance from the transformed sigma points.

In the context of angles-only orbit determination, sigma points are chosen from the measurement space (angles), and are then run through the orbit determination process. After running the sigma points through the AIOD process, a new mean and covariance are calculated from the transformed sigma points. The new mean and covariance can then be propagated to each of the measurement times to better estimate the effects of noise on the orbit solution obtained. The resulting covariance lets us draw conclusions on the quality of the orbit solution reached. If the covariance is large, then the solution is not very good and vice versa. Using the initial covariance found by applying the UT, sequential filtering techniques can be applied to propagate the mean and covariance forward from the measurement times.

In the following algorithmic pseudo-code, the Unscented Transform process is outlined. It should be noted that the original notation of Uhlmann will be followed, with application given to the AIOD problem (Julier et al., 2000).

Unscented Transform Pseudo-Code

1. Given items are n data points $\mathbf{x}(\mathbf{k})$ with an initial covariance $P(k|k)$, and mean $\hat{\mathbf{x}}(\mathbf{k}|\mathbf{k})$.
2. Approximate the given distribution with $2n+1$ weighted samples, χ_0 (sigma points).
3. Select sigma points with the algorithm, for $k=1$ to $2n+1$:

$$\chi_0(k|k) = \hat{x}(k|k) \tag{2.49}$$

$$W_0 = \frac{\kappa}{n+\kappa} \tag{2.50}$$

$$\chi_i(k|k) = \hat{x}(k|k) + \left(\sqrt{(n+\kappa)\mathbf{P}(k|k)}\right)_i \tag{2.51}$$

$$W_i = \frac{1}{2(n+\kappa))} \tag{2.52}$$

$$\chi_{i+n}(k|k) = \hat{x}(k|k) - \left(\sqrt{(n+\kappa)\mathbf{P}(k|k)}\right)_i \tag{2.53}$$

$$W_{i+n} = \frac{1}{2(n+\kappa)} \tag{2.54}$$

4. Transform each sigma point through the process model to give a set of samples in the desired space

$$\chi_i(k+1|k) = \mathbf{f}[\chi_i(k|k), \mathbf{u}(k), k]. \tag{2.55}$$

5. Compute the predicted mean as:

$$\hat{x}(k+1|k) = \sum_{i=0}^{2n} W_i \chi_i(k+1|k). \tag{2.56}$$

6. Compute the predicted covariance as:

$$\mathbf{P}(k+1|k)=\sum_{i=0}^{2n}W_i\chi_i(k+1|k)-\hat{x}(k+1|k)\times\chi_i(k+1|k)-\hat{x}(k+1|k)^T \qquad (2.57)$$

7. Now the transformed mean $\hat{x}(k)$ and covariance $\mathbf{P}(\mathbf{k})$ have been determined.

Uhlmann showed that the mean and covariance found through the UT accurately reflect the mean and covariance one would find through other processes, such as Monte Carlo simulations (Julier et al., 2000). However, the UT has the advantage of allowing the mean and covariance to be estimated with much lower computational cost then running Monte Carlo trials. The Unscented Kalman Filter (UKF) gives a more general approach to the Kalman Filter for estimating the covariance of Gaussian distributions and has been successfully applied to several areas in space mechanics, notably in the areas of spacecraft attitude estimation (Crassidis & Markley, 2003; Kim, Crassidis, Cheng, M. Fosbury, & Junkins, 2007). The UKF has also been shown to overcome certain flaws in the other versions of the Kalman Filter such as the Extended Kalman Filter (EKF). Because the UKF uses the complete non-linear system transformation for the mean and covariance, errors involving the first order linear approximation of the EKF are avoided, leading to a better solution (Wan & Van Der Merwe, 2000).

2.8.7 Kalman Filtering

Filtering is required for propagating the AIOD solution forward and to incorporate new observations into the orbit solution. Because the orbit solution state-vector contains 6 variables (i.e. position, velocity), 6 variables must be propagated forward along with a 6 × 6 covariance matrix that must be estimated.

The Kalman Filter (KF) is a well known filter that can be applied to the AIOD problem by propagating an initial orbit state mean and covariance forward in time. The advantage of the Kalman Filter is that new measurements can be incorporated along the way, allowing the covariance of the propagated states to be reduced when new measurements are added. The Kalman Filter operates on the basis of two main steps: a measurement step and a propagation step. The measurement step updates the Kalman Gain parameter by incorporating new measurement data. The measurement step then propagates the system model and covariance forward until the next measurement is available. In this way, an orbit can be propagated forward by initializing the filter with an initial orbit estimate and covariance and continuing forward through all available measurement points.

Various extensions to the original Kalman Filter provide improved results or address specific problems of the original. The Extended Kalman Filter (EKF) allows the use of non-linear functions for the estimation process, while the Unscented Kalman Filter uses an Unscented Transform to allow the use of any arbitrary full non-linear propagation function.

This concludes the discussion on background material and history of the orbit determination process. In chapter 3, the image processing and noise reduction techniques applied to the AIOD process for this research will be detailed. Definitions for the orbit determination process using angles-only have been presented in this chapter, as well as common noise sources.

3. Theoretical Development

Now that the necessary background information for orbit determination has been presented, the following sections provide details on the processes followed for using a small aperture optical system to provide the image data for orbit determination. First, image processing techniques used to prepare the image data for use in the orbit solution are presented. This is followed by discussions on techniques for object identification in images, including a method of discriminating candidate RSOs from a single long exposure image. Also discussed is a new "observability metric" for determining optimal sets of observations to use for the AIOD process. The chapter concludes with descriptions of an AIOD simulation tool developed for this research and the experimental setup used.

3.1 Image Processing

Images represent the data used for this research. Mathematically, images are a matrix of intensities $I(x,y)$, where the value at each pixel coordinate (x,y) gives a measure of the number of photons collected by that pixel. Because resident space objects (satellites) are generally quite dim, long exposures are usually necessary to reach a high enough signal to noise ratio (SNR) to distinguish them from the image background. Long-exposure images allow the collection of more photons by continuously collecting light for the duration of the exposure time. Unfortunately, as exposure times increase, different image noise factors such as read-noise and thermal-noise increase the spurious signals in the image.

The goal of applying image processing to space images is to improve the quality of the orbit solution from the measured position of RSOs in an image. Decreasing the noise level in images is an important part of accurately discriminating objects in an image.

Improving the accuracy of the point localization process should improve our resulting orbit estimate.

3.2 Noise Reduction

The purpose of noise reduction methods is to increase the signal to noise ratio of an image. Figure 3.1 shows a very poor quality space image with high noise levels and image artifacts due to camera defects. The removal of random shot noise is the main subject of the following sections. To remove the random noise there are several techniques that can be applied including image stacking, image calibration, and image filtering.

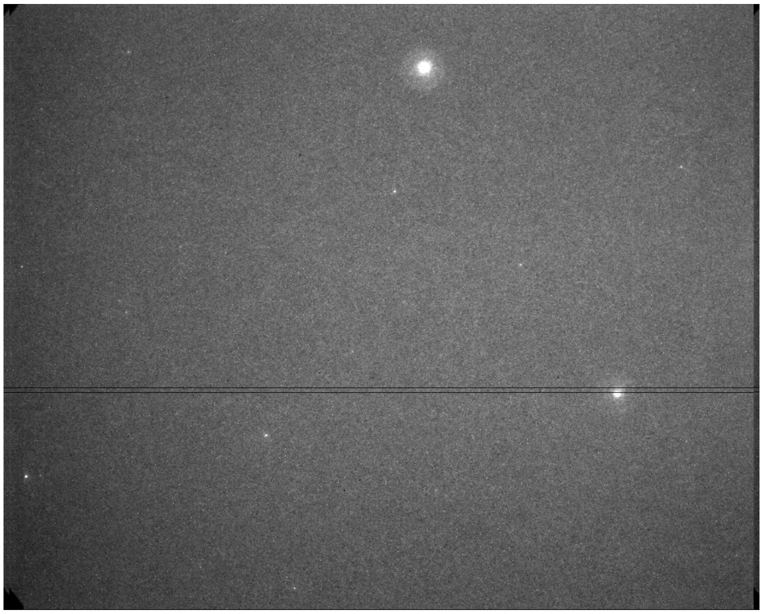


Figure 3.1. Noisy Image Example

The image shown in Figure 3.1 is a huge departure from the perfect ideal image. Given a perfect space image, the only intensities captured would be the light from a star, stellar object, or RSO while the image background would remain perfectly black. However, because of thermal effects, electronic noise, atmospheric effects, and optical distortions, perfect images are not possible. The goal of noise reduction is to bring an image such as that shown in Figure 3.1 to be as close as possible to the theoretical perfect image. In the following discussion of image processing, the data image will be referred to as a "light frame."

3.2.1 Stacking Images

Given a set of contiguous space images (contiguous images are images that capture the same field of view), the effects of random noise can be reduced by "stacking" a series of images. If we are taking space images with star fields that are the same, then the star field will remain fixed between frames if we track the motion of the stars. That is, the telescope and camera track the sky at the rate the earth rotates thus keeping the stars stationary from image to image.

While of great help for improving the SNR of static objects in an image such as stars or galaxies, stacking images containing RSOs often leads to a lower SNR for the RSO. For the case of stationary images, any RSO streaking through the image comes out with a very low SNR due to the fact that the pixels occupied by the RSO are only occupied for a short time, and never on subsequent frames due to the RSOs motion. When stacking, the RSO may also be missing entirely from the final image. The one case where stacking may be beneficial for RSOs is when using RSO tracking mode (section 2.4.2). In this mode, the RSO remains stationary between frames allowing the RSO signal to noise ratio to increase with stacking, while that of the background stars will decrease. However, without exact alignment between frames, the stacking process will often introduce an error around the RSO when the pixel locations are averaged together.

3.2.2 Dark Frame Subtraction

An effective means for removing random noise from images, as well as hot pixels is the subtraction of “dark frames.” A dark frame is an image taken with the same camera settings, temperature, exposure time, and gain as a light image. However, the dark frame is captured with the camera or telescope covered to block all light from entering (hence the name dark frame). The principle behind dark frames is to capture thermal effects, permanent banding, and dead pixels, which occur in all images. A set of dark frames is usually taken and one master dark frame created from an a stack of the entire group. This master dark frame is then subtracted from the light frame to remove some of the effects of random noise and any dead or hot pixels. Figure 3.2 shows a cropped portion of a master dark frame created from a set of 25 × 20 second exposures.

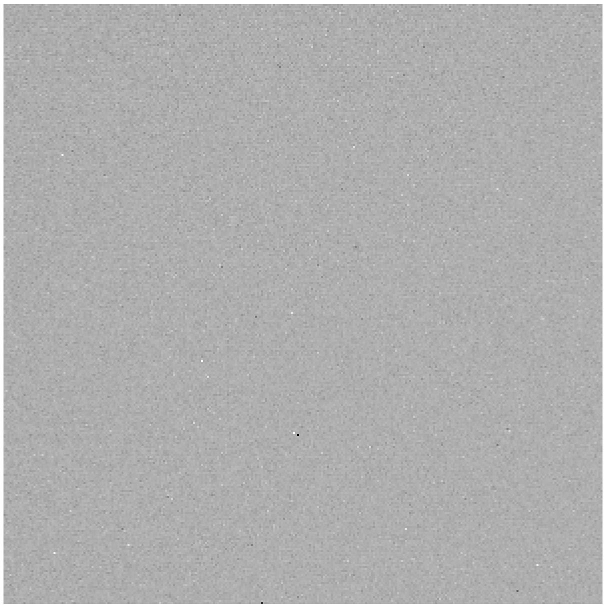


Figure 3.2. Cropped Portion of Dark Frame

3.2.3 Flat Frames

Flat frames are image calibration frames that correct for differences in the illumination of field. Flat frames can also correct for spots on an optical lens, or dust collection. Flat frames are images taken with all the same settings as a light image except exposure time. An evenly illuminated surface, such as an LED flat panel is often used to take flat frames. The exposure time is adjusted so that most of the camera's dynamic range is in use. Figure 3.3 shows a typical flat frame. Note that for this particular image there is a dust spot in the lower part of the image and the darkening of the image towards the corners, typical effects for telescope optics that do not have a perfectly illuminated field. Dividing the light image by the flat frame corrects for these differing illumination effects.

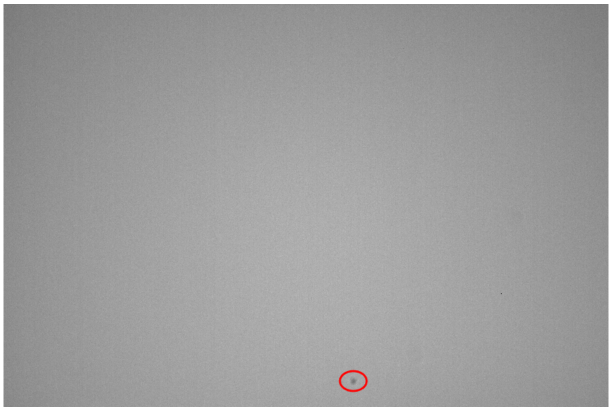


Figure 3.3. Flat Frame Example with Dust Spot

3.2.4 Bias Frames

Bias frames are created by capturing a camera's shortest possible exposure. Bias frames are usually taken with the aperture of the camera or telescope covered so as to not let stray light in, similar to dark frames. By taking the shortest possible exposure, bias frames capture the electronic read noise and damaged pixels of a camera quite effectively. As with other calibration frames, it is desirable to create a large set of Bias images and average the frames to form a master offset/bias frame. Bias frames are then subtracted from the light image, as well as being used to calibrate each individual dark and flat frame. Note that all calibration frames should be captured at the same temperature as the light frames they are used to calibrate.

3.2.5 Image Calibration Process

The process of applying Dark Frames, Flat Frames, and Bias Frames to images is called image calibration. The goal of image calibration is to reduce the overall noise level of an image as well as correct for optical and camera defects. Figure 3.4 shows the overall image calibration process given Dark, Bias, and Flat calibration frames.

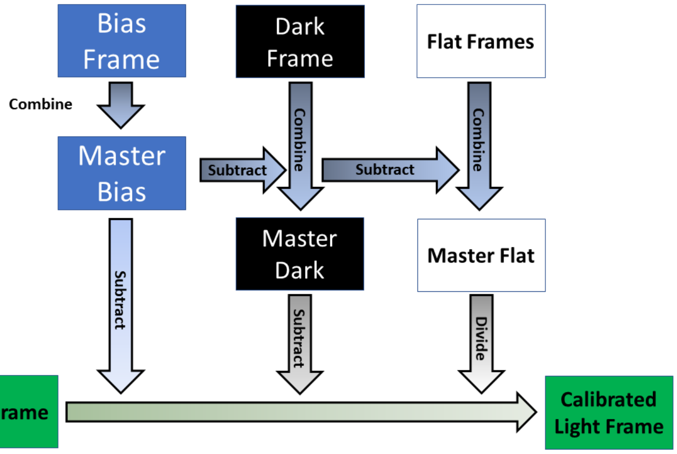


Figure 3.4. Image Calibration Process Flow Diagram

3.2.6 Noise Filtering Methods

Another method of reducing image noise is the application of image filters. Several well known techniques include median filtering, Gaussian smoothing, and mean filtering. Image filters are applied at the pixel level of an image. Noise reduction filters are usually applied after any image calibration steps are taken.

Median Filter

Median filtering works by looking at pixel areas and finding local medians. Then if the pixel values change drastically between pixels, this is usually a sign of noise if the pixel area under question is very small. Therefore, taking the median value of an *nxn* area of pixels, the original image can be corrected to better represent the distribution of nearby pixels.

Gaussian Smoothing Filter

A Gaussian smoothing filter works by suppressing high-frequency content in the image. Smoothing is accomplished by defining a Gaussian kernel operator $G(x,y,\sigma)$ and taking the convolution of this kernel with the image intensity map (Sease, Flewelling, & Black, 2017). The image is blurred by the convolution with the Gaussian kernel, which attempts to apply a Gaussian point spread function (PSF) to the image. Each pixel in the smoothed image is a weighted average of its neighboring pixels, with the mean value weighted towards the central pixels (which keeps the image in a Gaussian distribution). Gaussian filtering acts as a type of low-pass filter and removes high-frequency content from the image (high-frequency image content is usually random image noise). Equation 3.1 gives the form of the Gaussian Kernel operator where σ is the standard deviation chosen for the assumed distribution. Taking the convolution of the Gaussian Kernel and the image then gives the smoothed image as shown by equation 3.2. Note that "$*$" denotes the convolution operation.

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.1}$$

$$I_{smoothed}(x,y) = G(x,y,\sigma) * I(x,y) \tag{3.2}$$

Threshold Filtering

Another important filtering approach for space images is the threshold filter. Threshold filtering is usually applied after a light image has been calibrated and/or had a smoothing filter applied. Remaining background noise in an image can be removed by defining a minimum pixel intensity threshold. The threshold can be chosen from a point on the image histogram, or based on the mean background value of the image. Equations 3.3 and 3.4 show the process of finding a threshold value and then setting all pixels below this value to zero.

$$threshold = mean(I(x,y)) + N \times std(I(x,y)) \tag{3.3}$$

$$I(x,y) < threshold = 0 \tag{3.4}$$

Example of Image Calibration and Filtering

Figure 3.5 shows the results of applying image calibration, Gaussian filtering, and threshold filtering to a raw space image. The final image clearly shows streaks, which are GEO satellites, moving through the static star field. Notice how the streaks are barely visible in the original image but are more visible in the processed images. In figure 3.5, a progression of the calibration steps is shown. Step (a) shows the raw image, while step (b) shows the calibrated image. Step (c) shows the calibrated image with Gaussian smoothing applied. And finally, step (d) shows the result of applying a threshold filter.

3.3 Object Detection, Localization, and Discrimination

A key step in detecting objects in a space image is a proper application of noise reduction to the image before beginning the detection process. In this section, RSO de-

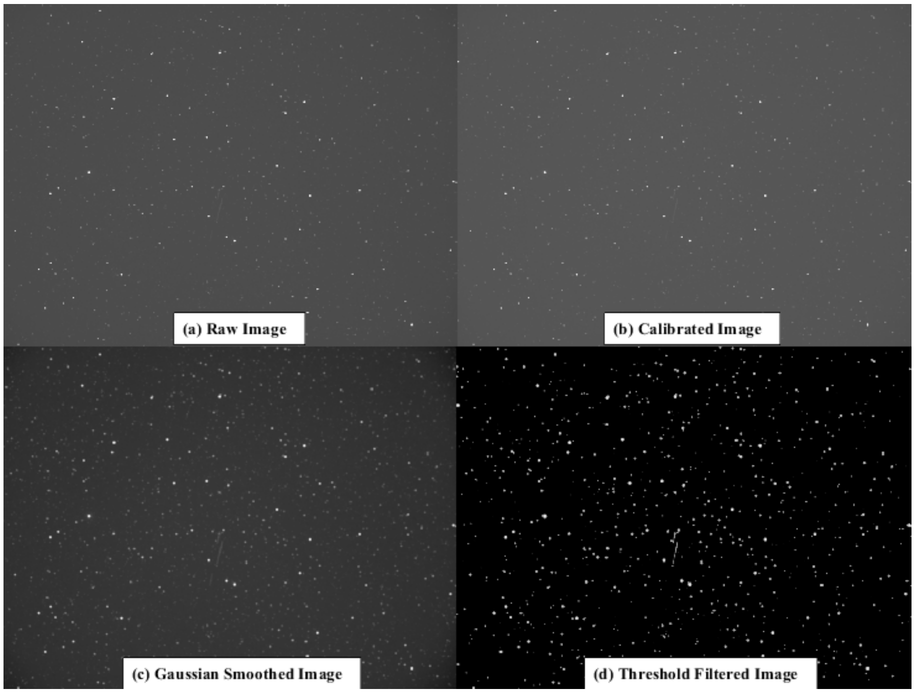


Figure 3.5. Image Calibration Example

tection methods for single images and discrimination across images are discussed. Some methods to find objects include centroiding, corner detection, streak detection, and gross motion analysis. Other techniques not on this list also exist.

3.3.1 Centroiding

In the process of finding possible RSOs in an image, the first step is to identify all objects in a given image. This is done by finding the centroids of regions of contiguous pixels in the image. The process of separating the objects in an image from the background sky is often referred to as "segmenting" the image. There are various algorithms for finding object centroids, however, an intensity weighted method was the primary method employed by this research. The centroiding process is usually performed after the noise reduction and image calibration steps to maximize the likelihood of successfully identifying objects. Typical centroiding techniques use an intensity-weighted method for finding the centroid of an object, and are capable of sub-pixel accuracy (Mortari, Bruccoleri, La Rosa, & Junkins, 2002). Given a set of N contiguous pixels, the intensity weighted centroid of an object $(\mathbf{x_c})$ can be found by equation 3.5.

$$\mathbf{x_c} = \frac{\sum_{i=1}^{N} \mathbf{x_i} w_i}{\sum_{i=1}^{N} w_i} \tag{3.5}$$

The pixel locations $\mathbf{x_i}$ correspond to pixel locations from a masked version of the image where every object has been separated from each adjacent object, and w_i is the intensity of the ith pixel. Such an image mask can be formed by using a threshold filter such as discussed in section 3.2.6. MATLAB's built in function *regionprops* was used to find the intensity weighted centroid locations of images. Tuning the threshold level affects the number of objects detected.

An example of applying this centroiding method can be seen in Figure 3.6. While it is feasible to find the centroid of every object in an image, this can be computationally inefficient, and results in finding the centroids of any remaining random noise pixels. To overcome this, a criteria for the minimum allowable area of a contiguous pixel region is set. Any objects in an image not meeting this requirement are removed from consideration.

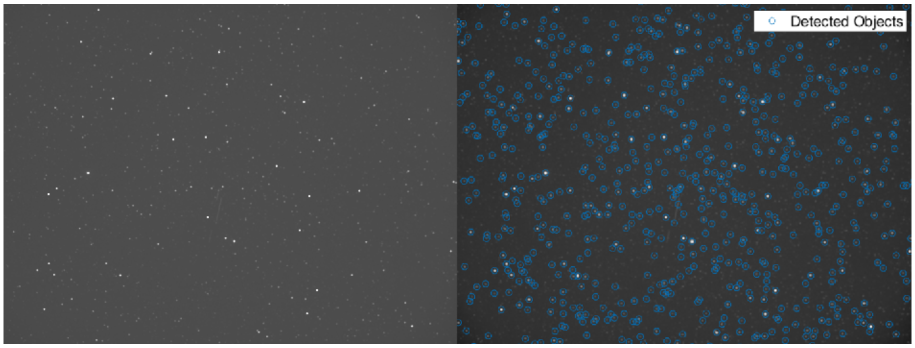


Figure 3.6. Detected Centroids From Processed Image

Once we have identified all the objects in an image, these objects must then be labeled as either stars or RSOs, the definition of which will depend on the type of image being processed. For RSO tracking mode images (see section 2.4.2), the stars are all streaks and the RSOs appear as points. However, if the image is taken in the sidereal tracking mode (see section 2.4.1), then stars will appear as point sources and any RSOs will show up as streaks. Using this knowledge a logical process can be followed to find candidate RSOs in either type of image. The discrimination process for objects is further discussed in section 3.4.

3.3.2 Corner Detection Methods

Detection of corners is one method for finding streaks in images. If the space images being processed have RSOs that streak through the image (i.e. sidereal rate tracking), then finding corners that are widely spaced in an image will locate the streak in the image. Corner detection methods begin by taking the first and second spatial derivatives of the image. An approximation of the spatial derivatives can be found by using the operators K_x, and K_y, which are commonly referred to as the Prewitt operators. Convolving the Prewitt operators with the image results in the approximate spatial derivative.

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.6}$$

$$K_y = K_x^T \tag{3.7}$$

Note that the Prewitt operators are among the simplest ways to find spatial derivatives and work by performing a finite difference approximation of the derivatives. If we take $I(x,y)$ to be the intensity value of an image at coordinates x,y, then the spatial derivatives are given by the convolutions of the image with the Prewitt operators. This process is shown by equations 3.8 and 3.9.

$$\frac{\delta I(x,y)}{\delta x} \approx K_x * I(x,y) \tag{3.8}$$

$$\frac{\delta I(x,y)}{\delta y} \approx K_y * I(x,y) \tag{3.9}$$

Since the Prewitt operators are essentially a first order derivative approximation, they are quite vulnerable to image noise. To mitigate the effects of random image noise on finding the spatial derivatives, a Gaussian smoothing filter is usually applied to the image before finding the spatial derivatives. Gaussian smoothing of an image can be accomplished by using a Gaussian kernel $G(x,y,\sigma)$ and a convolution with the image intensity map $I(x,y)$, a process discussed in section 3.2.6.

$$I_{smoothed} = G(x,y,\sigma) * I(x,y) \tag{3.10}$$

Figure 3.7 shows an example of the first spatial derivative in the x-direction of a space image with stars streaking through. The stars streaking through the image are clearly visible because of the strong pixel gradients created against the background intensity level. The

A fundamental method for finding corners in an image is known as the *minimum eigenvalue method*, which requires the second spatial derivatives which are found with equations 3.11 and 3.12 (Sease et al., 2017). Corners are identified by searching through the eigenvalues of a structural tensor formed from the spatial derivatives of the image. Equation 3.13, gives the image tensor.

Figure 3.7. Image Gradient Map

$$\frac{\delta^2 I(x,y)}{\delta x^2} \approx K_x * \frac{\delta I(x,y)}{\delta x} \tag{3.11}$$

$$\frac{\delta^2 I(x,y)}{\delta y^2} \approx K_y * \frac{\delta I(x,y)}{\delta y} \tag{3.12}$$

$$Z(x,y) = \begin{bmatrix} \left(\frac{\delta^2 I(x,y)}{\delta x^2}\right)^2 & \frac{\delta I(x,y)}{\delta x}\frac{\delta I(x,y)}{\delta y} \\ \frac{\delta I(x,y)}{\delta y} & \left(\frac{\delta^2 I(x,y)}{\delta y^2}\right)^2 \end{bmatrix} \tag{3.13}$$

Examining the eigenvalues of each 2×2 element of $Z(x,y)$ then gives an idea as to the likelihood that a corner exists at a particular (x,y) location. The structural tensor $Z(x,y)$ gives information on the gradients at every (x,y) location in the image. The gradients then give a measure of what each pixel location represents, whether that be a corner, flat area, or edge. If both of the gradients are large, then the (x,y) location represents a corner. Note that searching an entire image and finding all the locations of strong (x,y) gradients is computationally expensive. However, an exhaustive search for corners in an image can be mitigated by only searching around the pixel regions already known to contain an object, information readily available from object centroiding as discussed in section 3.3.1.

A “cornerness metric”, can be defined for the minimum eigenvalue method by finding the minimum eigenvalues of each 2×2 element in $Z(x,y)$. Using the minimum of the two eigenvalues ensures that $c(x,y)$ is large only if both of the eigenvalues are large.

$$c(x,y) = min\left(\lambda_1(Z(x,y)), \lambda_2(Z(x,y))\right) \tag{3.14}$$

Note that “large” is a relative term for an image intensity map and may require tuning of a threshold parameter to correctly identify corners in an image (Shi & Tomasi, 1994; Sease et al., 2017). Figure 3.8 shows the result of applying the minimum eigenvalue method to find corners in a processed space image. Note that corners were found on many objects, mostly stars, and not on the streaked RSOs that are in the center of the image.

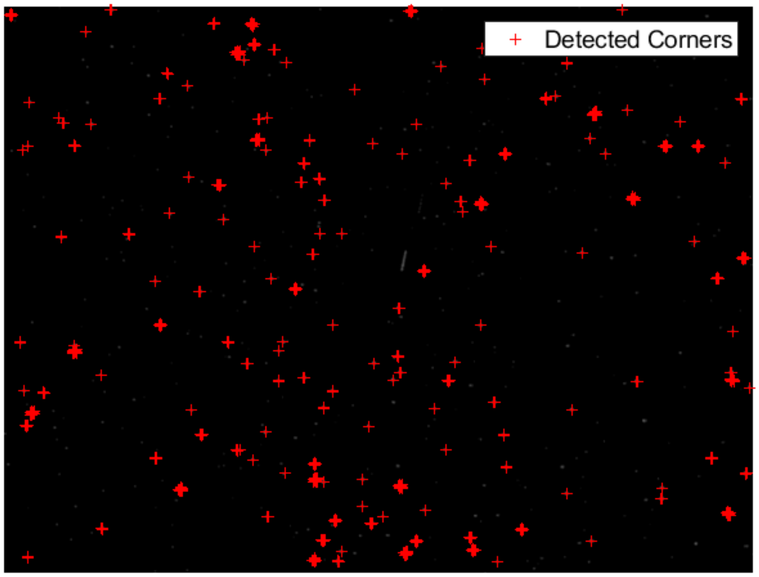


Figure 3.8. Minimum Eigenvalue Detected Corners

3.3.3 Harris Corner Detection

Harris Corner detection is a well known algorithm for finding corners in an image. Strong bi-directional gradients occur at corners and can be found by using an approximation for the eigenvalues of $Z(x,y)$ (G. Harris & J. Stephens, 1988). The Harris corner detector is computationally lighter than the minimum eigenvalue method by virtue of the approximation used for the cornerness metric instead of an explicit solution for the eigenvalues (Sease et al., 2017). Equation 3.15 shows the approximation used by Harris to determine the likelihood that (x,y) contains a corner.

$$c_h(x,y) = \left(\frac{\delta^2 I(x,y)}{\delta x^2}\frac{\delta^2 I(x,y)}{\delta y^2}\right)^2 - \left(\frac{\delta I(x,y)}{\delta x}\frac{\delta I(x,y)}{\delta y}\right)^2 - k\left(\frac{\delta^2 I(x,y)}{\delta x^2} + \frac{\delta^2 I(x,y)}{\delta y^2}\right)^2 \tag{3.15}$$

Figure 3.9 shows the results of applying the Harris Corner detection method to an image taken in RSO tracking mode. The 200 strongest corners have been plotted and can be used to identify the endpoints of the stars that streak through the image. If applied to an image with stars stationary, and an RSO streak, then the Harris Corner detector can be used to localize the endpoints of the RSO streak. MATLAB's built in function *detectHarrisFeatures* was used.

The Harris corner detector is a fundamental method for automatically finding streak endpoints in an image. By checking the distance between respective corners, and whether or not a contiguous object exists on either side of the corners, we can find the respective corner coordinates of a streak. If the image in question is a sidereal tracked image (see section 2.4.1), then finding a streak in an image means that the streak is very likely an

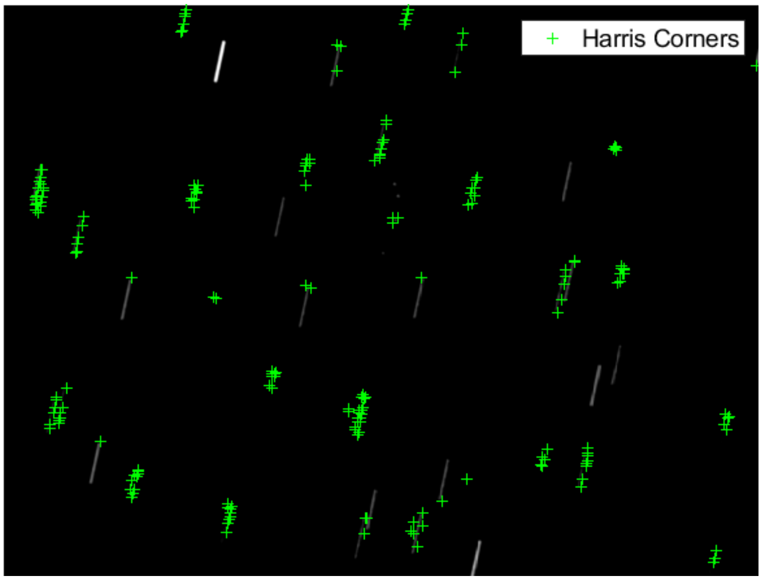


Figure 3.9. Harris Corner Detection Example

RSO. Brad Sease, et. al. used a similar method to automatically localize streak endpoints using the cornerness metric (Sease et al., 2017).

3.3.4 Line Identification Via Hough Transform

Streaks can also be identified in an image by finding line segments in the image. The Hough transform, which uses parametric representations of line segments, is used to find line segments in the image (Ballard & Brown, 1982). The Hough Transform is designed to produce candidate line segments from a reduced number of points in the image based on certain thresholding parameters. Adjusting these threshold can increase, or decrease the number of line segments detected. However, since the desire is to find obvious line segments in the image with the Hough Transform, the thresholds need not be set very low. Figure 3.10 shows an image captured by the author with line segments identified by the Hough Transform. The image was taken using RSO tracking mode, and all the streaks represent stars in the image. Note that not all the star streaks were found and labeled because of the threshold for what is considered a line by the Hough Transform.

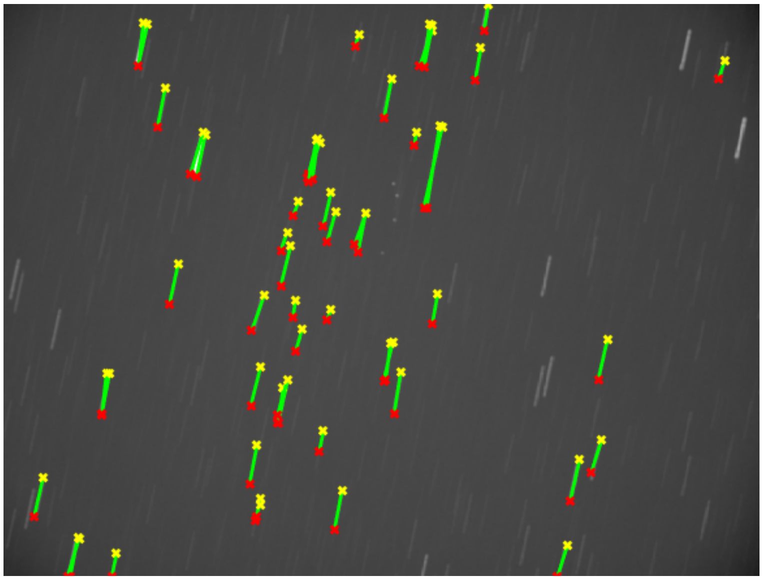


Figure 3.10. Lines Found with Hough Transform

3.3.5 Image Gradients

Image gradients are in important tool for finding objects in images. Mathematically, the gradient is the vector of partial derivatives of the image in the x and y directions respectively. For gray scale images, $I(x,y)$ is 2D and we can find the image gradients by using equation 3.16.

$$\nabla I(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix} \tag{3.16}$$

The magnitude of the gradient tells at what rate the intensity map $I(x,y)$ is changing, and the direction of the gradient tells the direction of this maximum increase. Figure 3.11 shows a visual representation of the 2D gradient of an image. The arrows represent the direction of maximum change of the intensity map, while the darker the shade, the large the magnitude of that change. The magnitude of the gradient at each (x,y) location can be found by 3.17, while the direction of the gradient can be calculated by equation 3.18.

$$g_{mag} = \sqrt{g_x^2 + g_y^2} \tag{3.17}$$

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right) \tag{3.18}$$

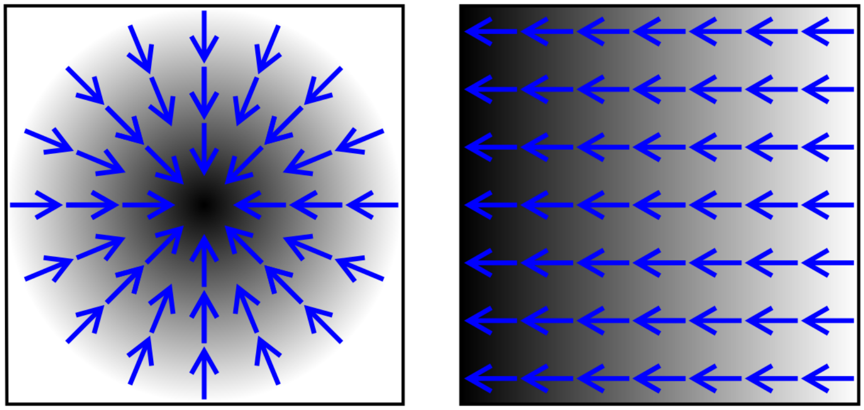


Figure 3.11. Image Gradient Example (“Image gradient”, 2018)

3.3.6 Local Gradient Search to Determine Object Type

While the Hough Transform and corner detection methods can be used to find lines, and corners of objects, further processing is then required to determine what type of object is represented by the line or collection of corners. Therefore, it is desirable to have a method of object discrimination that works as well or better than these methods for object discrimination, with the added ability of determining the type of object. In this section, a method of searching the local gradient around a detected object to arbitrarily discriminate the object as a line or point-source is presented. The number of detected point-sources and lines can then be used to determine which objects are stars and which objects represent candidate RSOs.

The method proposed searches the local gradient magnitude around detected objects looking for the characteristic gradient behavior of edges. The distance between strong edges is taken as the discriminating factor to label an object as a line or a point-source. The most efficient direction to search the gradient should be along the major axis of the current object. Therefore, having the object's major axis orientation angle is considered to be available *a priori* information.

Method Overview

The method begins by partitioning out a sub-image around the location of a detected object that has not previously been labeled as a line. The gradient of this sub-image is then calculated and examined to determine the type of object present. Figure 3.12 shows a colored representation of the gradient map around a typical streak object. Note the low gradient along the major axis until the edge is reached. Figure 3.13 shows a plot of the gradient magnitude along one direction of the major axis of the streak. Note the relatively low gradient level until the edge of the streak is reached. This distance is taken and compared against a length threshold.

Knowing the typical behavior of the gradient of an object allows us to determine whether the object is a line or point-source. Figure 3.14 shows the gradient map of a typical point-source object. (An image of the satellite Intelsat30 captured by the author in 2019.) Figure 3.15 shows the typical behavior of the gradient magnitude of a point source object. The main difference from a line being, that the characteristic short distance to an edge occurs in all directions, and not just perpendicular to the major axis.

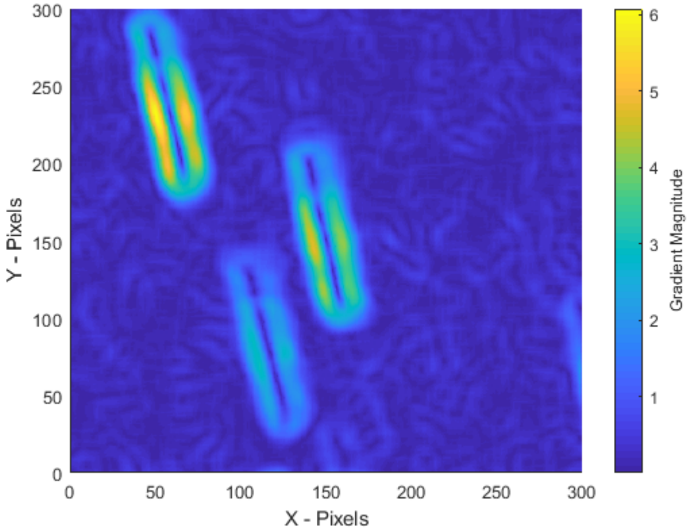


Figure 3.12. Streak Object Gradient Map

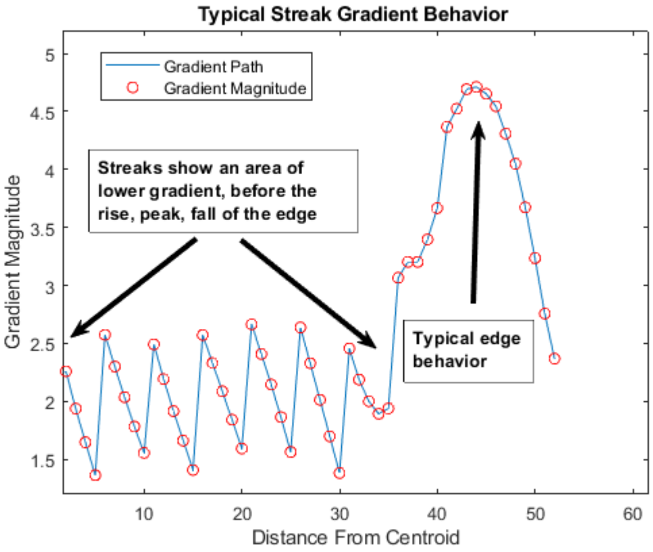


Figure 3.13. Streak Object Gradient Behavior

Now that the general behavior of the gradients of lines and point-sources has been discussed, a more detailed description of the object identification routine is presented in the following algorithmic pseudo-code.

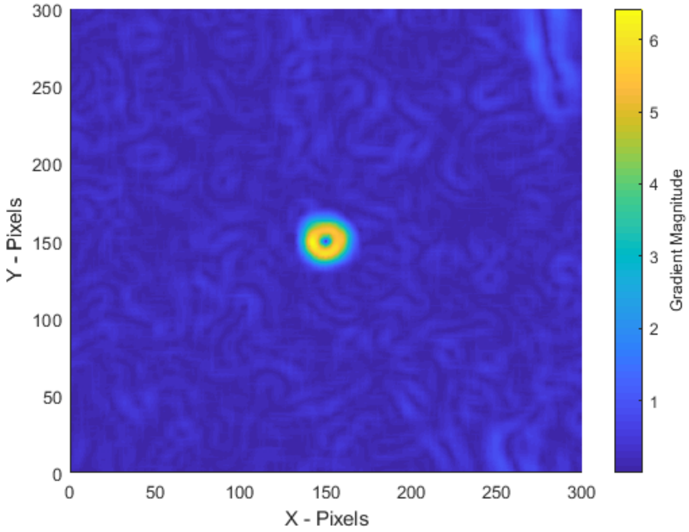


Figure 3.14. Point Source Object Gradient Map

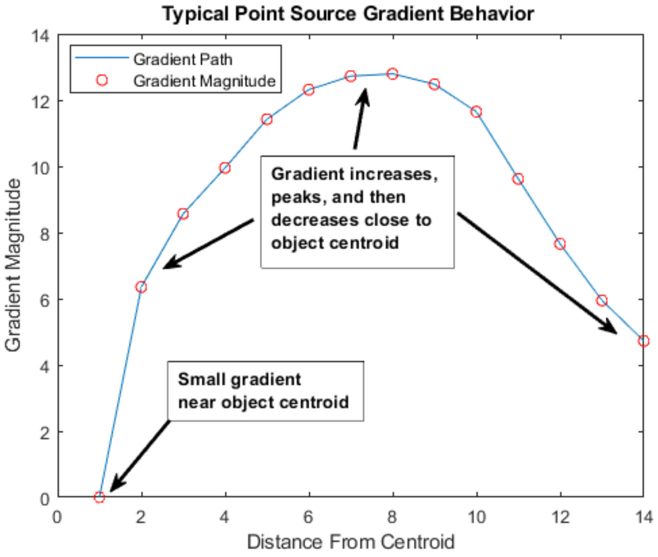


Figure 3.15. Point Source Object Gradient Behavior

Local Gradient Search for Object Identification Algorithm

1. Perform localized gradient search for objects not labeled as lines.
2. Given items for local gradient search are: object labels, $centroid_{type}$, object centroid locations $\mathbf{x_{c,i}}$, object major axis orientations θ_i for $i = 1, 2, \ldots, N$, with N = number of centroids.

3. For number of centroids not labeled as "*lines*", loop:

   (1) Partition out a sub image $I_{sub}(x,y)$ around the centroid location $\mathbf{x_{c,i}}$.

   (2) Find sub image gradient magnitude $GmagI(x,y)$.

$$\nabla I_{sub}(x,y) = \begin{bmatrix} g_x \\ \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix} \tag{3.19}$$

$$GmagI(x,y) = \sqrt{g_x^2 + g_y^2} \tag{3.20}$$

   (3) Search $GmagI(x,y)$ along direction θ_i until an edge is reached. Length traveled from $\mathbf{x_{c,i}}$ to edge is half of the object's length.

   (4) Check length of object against threshold for an object being considered a line.

   - If $streak_{length} > threshold$, label object as *line*.
   - If $streak_{length} < threshold$, we possibly have a point-source object. To make sure current object is a point-source do:

     i. Search $GradI(x,y)$ along direction $-\theta_i$ until an edge is reached.

     ii. If $streak_{length}$ in both directions is $< threshold$, then label current object as a *Point-Source*.

     iii. If $streak_{length}$ in $-\theta$ direction is $> threshold$, then label current object as a *line*.

     iv. If none of these conditions are met, or if we reach the edge of the subimage in both directions before finding an edge, label object as *unknown*.

   (5) If no more objects to check, end loop.

4. Output updated $centroid_{type}$ list of object labels.

3.4 RSO Identification and Discrimination

Using the techniques discussed above both point-source, and streak objects can be identified in an image. Now comes the task of labeling detected objects as stars or possible resident space objects.

3.4.1 RSO Discrimination From a Single Image

While it is possible to use techniques to identify candidate RSOs from multiple images, such as image subtraction, or gross motion analysis, the following proposed method attempts RSO discrimination from a single space image. The RSO discrimination process used in this research begins by first finding all objects in an image via the centroiding method discussed in section 3.3.1. Once all objects in an image have been found, the next task is to label these objects as lines or point-sources. Any objects with centroids that correspond to the midpoint of lines found by the Hough transform (section 3.3.4) are labeled as lines. Note that given a typical image, the Hough transform does not find all the line segments present, and thus we need another way to classify the remaining objects.

Taking the remaining objects to be classified, a sub-window of the original image is partitioned for further processing. Taking the information we know from the centroiding process (which gives us the centroid location, and object orientation angle), a search along the major axis of the detected object's gradient map is performed. If the object is a streak, then the gradient magnitude along the major axis will be small until we reach the edge of the streak. Once we reach the edge of the object, then we can compare the object length found by the change in gradient, to a threshold set for the length of a streak. If the detected length is greater than the threshold, then the object is labeled as a line.

The same method is applied to label an object as an RSO. If the edge of the object is reached and the detected length is less than the length threshold, it is possible the object is an point source object. Note that sometimes the centroiding process produces streak centroids not centered on the streak. To overcome this problem, we switch directions and search in the opposite direction. If the length found by searching in this direction is also less than the threshold for a line, then the object is labeled as an RSO.

Note that the aforementioned process labels objects as lines or RSOs, which will work for images taken in RSO tracking mode. However, if the image set being examined is taken in sidereal tracking mode, then the labels can be reversed so that any lines detected are RSOs and any point source objects are labeled as stars.

3.4.2 Object Discrimination Algorithm

In order to automatically determine the type of image being processed, a logic statement series must be followed. It is hypothesized that if an image is captured in RSO Tracking mode, then the majority of objects in the frame will be lines with similar orientations and lengths. Any objects found in this type of image that correspond to a point-source can then be considered to be RSOs. Note that given further analysis we could also use the fact that most lines will be oriented in one direction to detect RSOs that are non-point source objects that streak through the image. The following Pseudo-Code algorithm outlines the main steps of the process developed for this research to discriminate RSO objects from a single image. The core of the algorithm is the gradient-based search method outlined in section 3.3.6.

Object Discrimination Algorithm Pseudo-Code

```
1. Given items are: Light image $I(x,y)$, Calibration frames $I_{calib}(x,y)$,
2. The first step is to perform image calibration to $I(x,y)$.
       $I_{calibrated}(x,y) = I(x,y) - I_{dark}(x,y) - I_{bias}(x,y)$    (3.21)
3. Apply Gaussian Smoothing filter to calibrated image.
       $I_{smoothed}(x,y) = I_{calibrated}(x,y) * G(x,y)$    (3.22)
4. Remove remaining background noise by applying image threshold filter.
       $I_n(x,y) = I_{smoothed}(x,y) > threshold$    (3.23)
5. Find location of all objects in image by running weighted centroiding algorithm. Use MATLAB's *regionprops*.
6. Find possible lines in image by using a Hough Transform. Label all object centroids on the center of detected lines as line objects.
7. Perform localized gradient search to label remaining objects as "lines", "Point-Source", or "Unknown" objects (section 3.3.6).
8. Compare number of detected *lines* to number of detected *point-source* objects.
       If $\#lines > \#point\text{-}sources$, THEN $point\text{-}source = RSOs$
       Else $lines = RSOs$
9. Output centroid data and updated labels for use in orbit determination process.
```

The RSO identification algorithm is a multifaceted approach to find RSOs in a single space image. The utilization of multiple object detection techniques (centroiding, Hough Transform, gradient search), helps improve the success rate of the RSO discrimination process. Often images taken containing RSOs are desired to be spaced as far apart in the sky as possible. This spacing makes detection of RSOs from a single image with a method such as that outlined above desirable.

3.4.3 RSO Discrimination via Gross Motion

One other method of finding RSOs in an image uses the aggregate motion of all objects across several images. Given a set of sequential images, tracking all objects from frame to frame makes it possible to determine which objects in a frame correspond to stars, and which to RSOs. However, implementing this method requires a way to successfully associate objects across the images. This association limits the allowable time between images, and also places limitations on the physical frame to frame motion between frames. Additionally, if image data is sparsely scattered from different locations across the sky, association may not be possible at all, pointing to the utility of RSO discrimination using only a single image.

3.5 Extended Object Image Processing

In addition to lines and point-source objects, there are situations where a satellite is large enough, or close enough that it is at least partially resolved in an image. Extended objects can be defined as objects in an image occupying more than a few pixels, not necessarily symmetrical, and possibly showing some amount of structure of a spacecraft. For angles-only orbit estimation, angular measurements correspond to a point source localized by a centroid as discussed in section 3.3.1. However, when processing RSOs that are not point sources, care must be taken to choose a point on the object to use as the measurement for the AIOD process. Choosing a point to run through the AIOD process from an extended object is a non-trivial process. Applying luminance based centroiding algorithms to find the overall centroid of an object is one approach, the result of which can be seen in Figure 3.16. However, if the object is sufficiently large and shows significant amounts of structure, then it is possible to apply other image segmentation techniques to choose a point for the AIOD process (Tiwari et al., 2019).

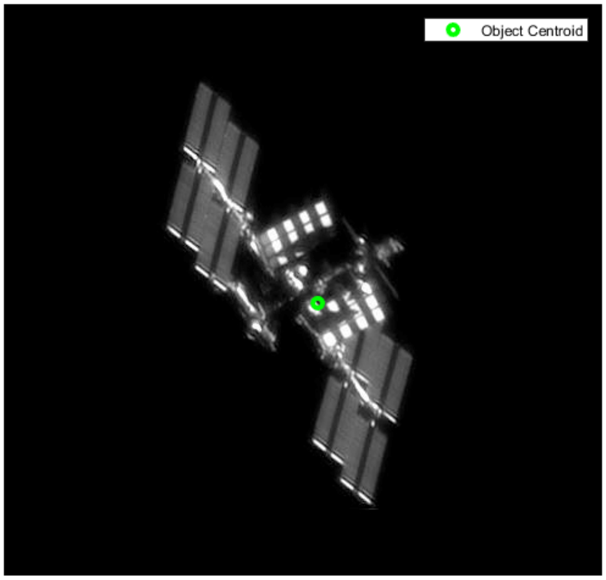


Figure 3.16. Extended Object Example

3.5.1 Template Matching (Extended Objects)

Given an image such as Figure 3.16, template matching can be used to identify the object in a series of images. Template matching uses the normalized cross-correlation matrix to determine areas on an image that correspond to a template image. Once the template has been found in an image, then the center of the object can be extracted based on the template location.

3.5.2 Feature Point Tracking

If template matching is too computationally expensive, or no template is available to match, a feature point tracking algorithm can be used to track points on an extended object from frame to frame (Tiwari et al., 2019). Figure 3.17 shows an image of the ISS with feature points matched from one frame to a reference frame. Once points have been chosen and tracked across frames, the orbit determination process can be completed if attitude information for each image is available. A full investigation of extended objects is a topic of future research.

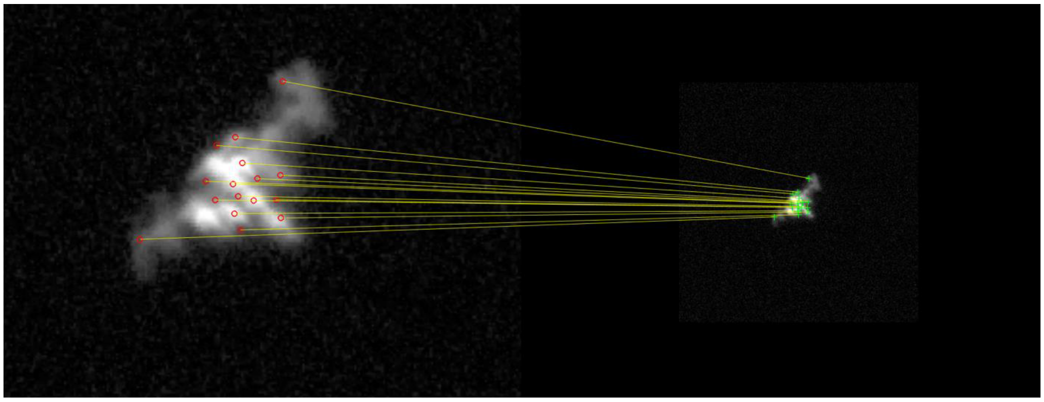


Figure 3.17. Matched Feature Points Between Frames (Tiwari et al., 2019)

3.6 Attitude Determination

Before the data from an image can be used for orbit-determination, the attitude of the camera must be known. Once we know the attitude of the camera, then we can find the inertial pointing direction to any detected objects in an image. There are several methods of obtaining the camera's attitude including the use of plate-solving, an onboard star catalog, onboard star matching (star-tracker), or if equipped and accurate enough, onboard ADCS solutions.

3.6.1 Plate Solving

The process of matching stars in an image to known star positions in a catalog is known as plate solving. Star Trackers use a similar process often with a limited catalog to speed processing. Lang et. al. have shown that a sufficient number of stars for matching is between 4 and 6 (Lang et al., 2010). Matching is done through a process of comparing star “asterisms” (an asterism is an arbitrary geometrical shape formed by a collection of stars) of known orientations to asterisms detected in the given star field. Astrometry.net is a computer program that will perform plate solving for any space image (Lang et al., 2010). An example of a successful plate solving operation from one of the images captured for this research is shown in Figure 3.18. The green lines connecting four bright stars shows the star pattern matched with the built in catalog of Astrometry.net.

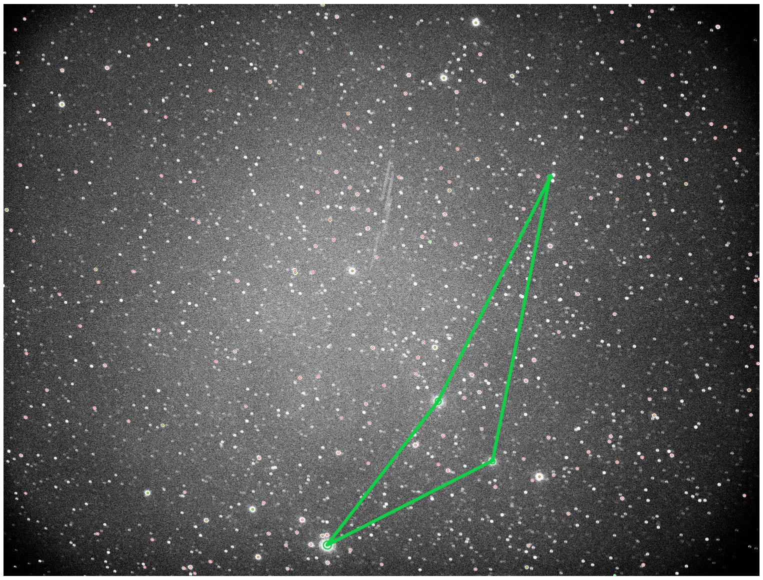


Figure 3.18. Astrometry.net Successful Plate Solution

3.6.2 Astrometry.net MATLAB Interface

A MATLAB interface was designed to use the astrometry.net plate solver. This allows astrometric data to be found directly from the use of MATLAB commands rather than utilizing any outside software. A standalone version of the astrometry application must first be installed in a CygWin envrionment to allow astrometry to run on Windows. An existing standalone installation of the astrometry.net solver is available from ANSVR (Galasso, n.d.).

Sending commands to astrometry.net is accomplished by using MATLAB to build a command line sequence to run the plate solve operation. (Note, astrometry.net will be referred to as astrometry from now on.) The commands used by astrometry have various options that affect the solution convergence and solution computation time. A summary of that in the author's experience most highly affect the convergence of astrometry to a successful solution are listed below.

Astrometry.net Options

1. "-z 4", Downsampling, options are 1x, 2x, 4x downsampling. This option downsamples the image before attempting to perform the plate matching process. When given large images, downsampling helps improve the solution time.
2. "-u" arcsecperpix -L $< lowvalue >$ -H $< highvalue >$. Image scale options, this is probably the most important factor in letting astrometry reach a solution. If the high and low values are left blank, astrometry will attempt to guess the pixel scale of the image. Pixel scale can easily be calculated though and passed in as a high and low value here.
3. "–crpix-center", While not an option that affects the convergence, this command tells astrometry to set the reference pixel for all coordinates to the center of the image.

There are many more options to adjust the plate solving parameters of astrometry, for more information please refer to the astrometry user's guide available online ("Astrometry.net code Readme", n.d.).

3.6.3 Onboard Star Catalog (General Case)

If the orbit determination process is to be done onboard for some mission, then it is desirable to use an onboard star catalog for finding image attitude rather than relying on a 3rd party software program. Star pattern matching can be accomplished by implementing a routine similar to a star tracker.

3.6.4 ADCS Solution

Another option for attaining attitude information can be from an onboard Attitude Determination and Control System. This option is especially attractive for space-based missions since most spacecraft are equipped with star trackers. Utilizing this information to find the camera boresight pointing direction could then be coupled with the camera's known physical location on the spacecraft to find the attitude of the image.

3.7 Conversion to Inertial Frame

Once an RSO has been successfully localized in the image frame, the resulting pixel location must then be transformed into a set of angular measurements for the AIOD process. The model chosen to use for this method is a simple pinhole camera model. The colinearity equations then allow us to write expressions for converting the pixel locations to a set of inertial angular measurements. The first step in converting a given centroid location $\mathbf{x_c} = \{x_c, y_c\}$ to an inertial measurement is to translate the pixel location to a relative position to the camera center (boresight). Equation 3.24 gives the expression for changing the pixel reference point.

$$\mathbf{x_c} = \mathbf{x_c} - \mathbf{x_{boresight}} \tag{3.24}$$

Now a unit vector in the camera frame can be formed by using the known camera pixel size (μ_{pixel}) and focal length (f) as shown in equation 3.25.

$$\mathbf{v_{cam}} = \begin{bmatrix} (x_c)(\mu_{pixel}) \\ (y_c)(\mu_{pixel}) \\ f \end{bmatrix} \tag{3.25}$$

Next convert the camera frame vector to a unit vector $\hat{\mathbf{v}}_{\mathbf{cam}}$.

$$\hat{\mathbf{v}}_{\mathbf{cam}} = \frac{\mathbf{v_{cam}}}{||\ \mathbf{v_{cam}}\ ||} \tag{3.26}$$

Given the rotation matrix of the camera bore sight, R_{cam}, found from attitude determination step (see section 3.6), we can now find the inertial LOS unit vector for the measured centroid location.

$$\widehat{\mathbf{L}}_{\mathbf{rso}} = R_{cam}\hat{\mathbf{v}}_{\mathbf{cam}} \tag{3.27}$$

Now that the inertial LOS vectors are available for detected RSOs, all the necessary information for running the orbit determination process has been found.

3.8 Initial Covariance

Once a set of measurements is available, an important next step is to develop an initial covariance estimate. This can be accomplished by using an Unscented Transform to generate sigma points to represent the mean and covariance at each measurement time. However, in order to do this, an attitude error estimate must be available.

In order to develop an initial attitude error estimate, we must have something to compare. Given the plate solving solution from astrometry, it is possible to use the known angular positions of stars to compute the attitude error of a space image. If we take the angular position of a few stars that have been detected in an image, the offset from known truth positions can be calculated. If this offset is calculated for a few stars in the image and averaged, this gives an idea as to the error in measurements.

Compute Measurement Covariance Pseudo-Code

```
1. Given items are: data image $I(x,y)_i$
2. Run object detection routine (section 3.4.2). Once an object of interest has been detected, find the inertial coordinates of the object.
3. Run plate solving operation to get inertial pointing direction of camera.
4. Convert all objects of interest from pixel frame to inertial (RA,DEC) frame (section 3.7).
5. Given star positions, $\{\alpha_i, \delta_i\}$, for $i = 1, 2, ..., N$ in RA, DEC coordinates from data image, compute offsets from stars in catalog.
6. Average the offset to arrive at an estimate of the standard deviation in RA and DEC, $\sigma_\alpha$ and $\sigma_\delta$ for the image.
7. Compute angular covariance matrix for image $I(x,y)_i$.

   $P_{meas} = \begin{bmatrix} \sigma_\alpha^2 & 0 \\ 0 & \sigma_\delta^2 \end{bmatrix}$   (3.28)
```

Once an initial covariance for the measurements has been computed, an Unscented Transform can then be used on each set of α_i, δ_i RSO coordinates to generate sigma points. For the regular three measurement AIOD case, this gives a total of 15 sigma points (counting the mean value), with 5 for each measured location. Running all combinations of these sigma points can then be used to compute an orbit estimate with an initial covariance.

While not affecting the actual orbit estimate, being able to compute the initial covariance tells about the quality of the estimate. Additionally, an initial covariance is required for the application of filtering to the orbit in order to propagate forward to each measurement time and beyond. Having the initial covariance is also essential for running precise orbit determination, batch filtering, or differential correction of the orbit.

3.9 Observability Metric

Given an arbitrary set of images to use for the AIOD process, we can now automatically find potential RSOs from any single image. However, we still need a way to choose which set of images to use for the AIOD process. The concept of observability from Control Theory can be used to develop a metric for gauging the quality of the LOS observations used for the AIOD process. (For a complete review of observability please see (Brogan, 1985), or (Lewis, Syrmos, & Vrabie, 2012)). A new method is proposed whereby the determinant of the LOS vectors is investigated in a manner similar to looking at the observability matrix of a system.

3.9.1 Observability Metric (3-Measurements)

From Linear Systems Theory, we know that the determinant of a matrix gives a measure of the linear independence of each respective column (Strang, 2003). It is then proposed that if we take a set of three LOS observations, concatenating them into one matrix, the determinant of this LOS matrix will give us a metric to gauge the quality of the set of observations for use in AIOD methods. Let $\widehat{\mathbf{L_i}}$ be any LOS observation vector obtained from an image containing an RSO. Taking any three observations in series, $\widehat{\mathbf{L_i}}$, for $i = 1, 2, ..., N$, form a 3×3 matrix, where $a > i$ and $b > a$ are the indices of the chosen measurements.

$$L_{obs} = \begin{bmatrix} \widehat{\mathbf{L_i}} & \widehat{\mathbf{L_{i+a}}} & \widehat{\mathbf{L_{i+b}}} \end{bmatrix} \tag{3.29}$$

The determinant of L_{obs} gives us information about how linearly dependent each of the different observation vectors are from one another. Given that all the observations will be nearly in the same plane (the orbit plane of the observed satellite), then the farther apart these observations are spaced, the larger the determinant of L_{obs} should be. However, we have no guarantees that this determinant will be positive, a problem easily overcome by examining the magnitude of the determinant. Let ψ_{obs}, as defined in equation 3.30, be the proposed metric for determining the best set of observations for use in the AIOD process.

$$\psi_{obs} = \mid det\left(L_{obs}\right) \mid \tag{3.30}$$

Given a small separation between measurements, we would expect the determinant to be rather small, say on the order of 10^{-8}. If however, our observations are more properly spaced, we would expect to see the value of the determinant become larger approaching some upper limit. A value of 1 would correspond to a set of perfectly orthogonal observations. Given the orbit problem geometry, all observations will be nearly in the same plane (the plane of the satellites orbit), therefore perfectly orthogonal measurements can never be expected.

3.9.2 N-Measurement Observability Metric

Extending the observability metric ψ_{obs} to cases with $N > 3$ measurements is an obvious next step. With $N > 3$ measurements, the matrix formed by the LOS vectors will no longer be square, making the determinant undefined. Under current investigation is examining the condition number or something similar to the condition number of the matrix formed by the LOS observations as shown in equations 3.31 and 3.32.

$$A = \left[\widehat{\mathbf{L}}_\mathbf{1} \widehat{\mathbf{L}}_\mathbf{2} \ldots \widehat{\mathbf{L}}_\mathbf{N} \right] \tag{3.31}$$

$$\Psi_N = cond\left(A\right) \tag{3.32}$$

A full extension of the observability metric ψ_{obs} to the case of $N > 3$ measurements is an item for future research.

3.10 Simulation Tool

In order to test the observability metric ψ_{obs} discussed in section 3.9, a MATLAB tool for simulating angular observations was developed. A MATLAB implementation of the Special General Perturbations 4 (SGP4) algorithm propagates a target satellite to desired observation times. Measured LOS vectors are generated by taking the vector difference from the site location and creating a unit vector. Adding Gaussian noise to the observation vectors allows the examination of the effect of noise on different orbit scenarios.

3.10.1 Propagation Methods for Generating LOS Measurements

Measurements to a target satellite are generated by propagating the state of the satellite from the epoch time forward to the time of observation. An initial satellite state vector is obtained from a Two-Line Element (TLE) file at a given epoch time. Next, the initial state is propagated forward to desired measurement times via SGP4. The observer's location is set as latitude, longitude, and altitude coordinates. Then the site coordinates are converted to a set of ECI position vectors at the time of each observation. Since the position vector to the site ($\mathbf{R_i}$) and the position vector to the satellite ($\mathbf{r_i}$) are known, the vector geometry defined in 3.33 allows the determination of the position vector to the satellite ($\rho_i\widehat{\mathbf{L_i}}$). The unit vector observation $\widehat{\mathbf{L_i}}$ can be formed by dividing by the norm of the satellites relative position vector as shown in equation 3.34.

$$\rho_i\widehat{\mathbf{L_i}} = \mathbf{R_i} - \mathbf{r_i} \tag{3.33}$$

$$\widehat{\mathbf{L_i}} = \frac{\rho_i\widehat{\mathbf{L_i}}}{||\ \rho_i\widehat{\mathbf{L_i}}\ ||} \tag{3.34}$$

3.10.2 Adding noise to generated LOS

To better represent a real measurement, noise is added to the observation unit vectors from a Gaussian distribution. Considering the LOS vector to be some mean value, letting noise be added to the LOS vector creates a 3D cone around the unit vector where anywhere in the cone can be a point from the Gaussian noise added.

3.11 Experimental Setup

While simulation of observations is important, nothing can replace using experimental data. This section provides some details about the experimental setup utilized for this research to capture space images.

3.11.1 Hardware

With the goal of using small aperture optical systems, the optical instrument used was an 11 inch Rowe Ackermann Schmidt Astrograph telescope (Berry & Celestron Engineering Team, 2016). The RASA telescope was designed specifically for wide, flat-field imaging of the night sky. This combination makes the instrument ideal for the task of space surveillance. The fast F2 optical system allows the use of short exposure times to capture dim targets. Given the high relative motion of satellites in the sky (except for

GEO objects), lower exposure times give the best chance for successfully capturing images of a satellite. Figure 3.19 shows the complete system ready to observe.

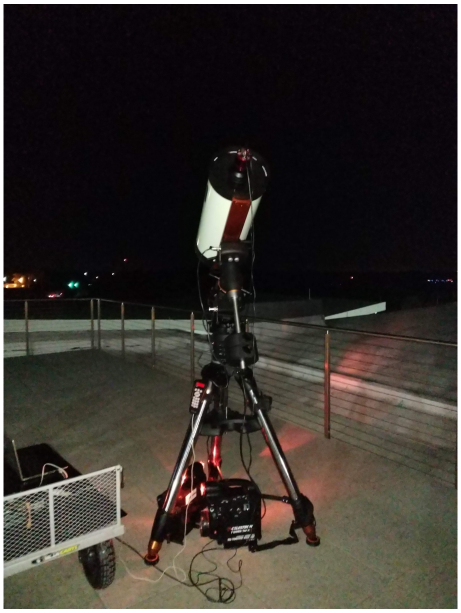


Figure 3.19. Experimental Setup

The telescope was mounted on a Celestron CGE-Pro equatorial tracking mount. With a payload capacity of 90lbs, and slew rates up to $5\frac{\text{deg}}{s}$, this mount provides a sturdy, accurate platform for imaging satellites. More information on the CGE-Pro mount can be found on the manufacturer's website: *https://www.celestron.com/products/cge-pro-mount*.

The camera used was an ASI1600M monochrome cooled camera which is a 16 megapixel cooled CMOS camera specifically designed for astrophotography. Detailed specifications of the camera can be found in the product manual and on the manufacturer's website (ZWO Company, 2016).

3.11.2 Software

Hardware is extremely important, but hardware runs through the use of software. In this section, a short description of how different software programs were used for implementing this research will be described. The goal of this section is to provide an overview of the software used by this research for those interested in work with similar systems.

MATLAB

MathWorks, Matrix Laboratory (MATLAB) software was used for all computational and image processing tasks in this research. Numerous functions and scripts were written using MATLAB to implement the algorithms presented in this research.

Astrometry.net

Astrometry.net is a computer software program that provides astrometric plate solving of arbitrary space images ("Astrometry.net", n.d.). Astrometry.net is written in a Linux distribution and for use in windows must use an interface program such as Cygwin to provide the Unix interface for Astrometry's functions. The implementation used by this research uses the astrometry interface provided by ANSVR (Galasso, n.d.).

ASCOM

A set of common standards and drivers called ASCOM provide a unified framework of drivers that allow computer interfacing with various pieces of astronomical equipment through standard windows COM objects. An ASCOM driver for your specific telescope can be downloaded from their website. This allows interfacing with the driver via a standard Windows COM object ("ASCOM - Standards for Astronomy", n.d.).

Astrophotography Tool (APT)

Astrophotography Tool (APT) is a telescope and imaging control software suite for Windows that allows camera and telescope mount control ("Astrophotography Tool", n.d.). The software was used to run the scripts that controlled the camera for all imaging sessions using the experimental setup described in section 3.11.

SatTracker

SatTracker is a windows application written to allow commercial telescopes to find and track satellites in the night sky (Eccles, 2015). SatTracker uses TLE data to propagate satellite positions and generate the commands necessary to track a satellite across the sky.

Stellarium

Stellarium is a planetarium software used to assist in planning observations. This software has the ability to display satellite positions in real time, or any specified time past or present. A plugin for Stellarium also exists that allows the user to control a telescope and send GoTo commands to a telescope through the ASCOM universal drivers ("Stellarium Astronomy Software", n.d.).

3.12 End-to-End AIOD Process

Now all of the necessary parts to run the AIOD process from capturing an image to finding the orbit have been defined. A summary of the entire process is presented in this section as a conclusion to the theoretical development. Emphasis is given to the techniques developed for this research. The end-to-end AIOD process begins with capturing a set of viable space images and concludes with an initial orbit estimate.

AIOD End-to-End Algorithm

1. Capture a set of N measurement images (I_n).
2. Segment all images for detection of RSOs (section 3.4).

3. Find observability metric for all combinations of measurements (section 3.9).
4. Choose measurement set corresponding based on observability metric.
5. Run orbit determination (section 2.2).
6. Examine Results.

3.13 Autonomous Orbit Determination and Tracking

All of the processes mentioned thus far for image processing and orbit determination have the capability of being automated. The future of space surveillance lies in automated survey and tracking systems. A few pieces of the puzzle still remain though before truly autonomous systems can be reality such as robust automatic RSO detection and onboard catalog creation.

4. Results

This chapter contains results from the application of the techniques discussed in chapter 3 to actual space images. All space images shown in this chapter were captured by the author using the experimental setup shown in section 3.11 and give a realistic expectation of what to expect from any similar small aperture system. Orbit estimation from simulated measurements provides a comparison point to the actual observations.

4.1 Image Processing Results

A data set consisting of images of the geostationary satellites INTELSAT30 and MEXSAT3 were captured from an observing location in Daytona Beach, Florida from January to February 2019. Image sets were captured in both RSO tracking mode and Sidereal rate tracking mode using a variety of exposure times. The goal of varying camera parameters was to provide a diverse set of images to attempt finding of candidate RSOs. Table 4.1 gives the orbital elements of the target satellites for the epoch nearest when the images were captured.

Both satellites are GEO satellites and are part of larger constellations, which means that images containing the target satellites also contain several other satellites. Multiple satellites in an image poses a great test for the object identification algorithm presented in section 3.4.2.

Both image sets consist of images taken in the ”RSO Tracking” mode. Multiple exposure times were used ranging from 5 seconds to 20 seconds. Camera gain settings were kept constant. The image sets span 70 minutes in 69 images for INTELSAT30, and

Table 4.1

Test Case Satellite Orbital Elements

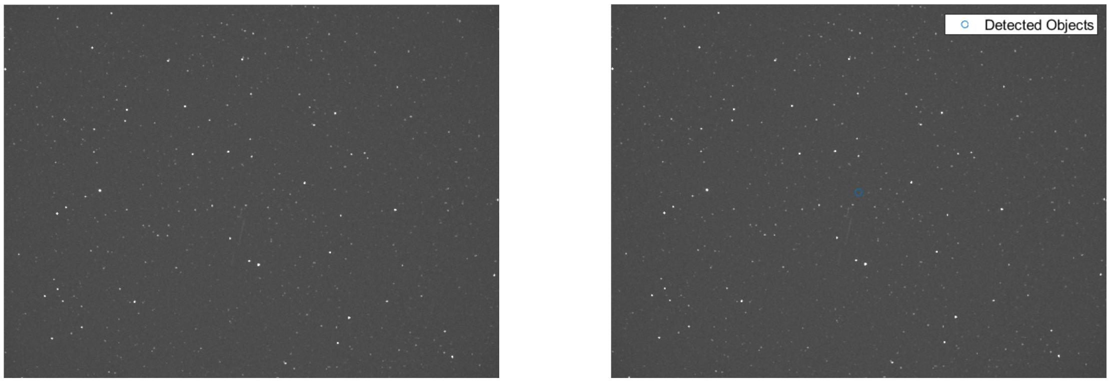
		Test Case A	Test Case B
Test Case Orbital Elements	Units	INTELSAT30 (GEO)	MEXSAT3 (GEO)
Epoch	JD	2458519.71	2458520.44
a	km	42164.65	42164.78
e	-	2.02E-04	1.88E-04
i	deg	0.0165	0.010
Ω	deg	72.80	124.74
ω	deg	46.15	195.66
M	deg	357.50	39.5581

50 minutes in 49 images for MEXSAT3. In both cases the arc of the orbit observed is relatively small for a GEO satellite.

4.1.1 Object Centroiding Results

Applying the noise reduction techniques discussed in section 3.2 improves object detection, and in most cases is absolutely essential to obtain any meaningful results from the experimental images. For example, take Figure 4.1, which shows side by side views of the raw image, and then centroids detected in the raw image. Without taking some steps to reduce the background noise, the centroiding algorithm is unable to segment any objects and thus the only detected object is a centroid corresponding to the center of the image.

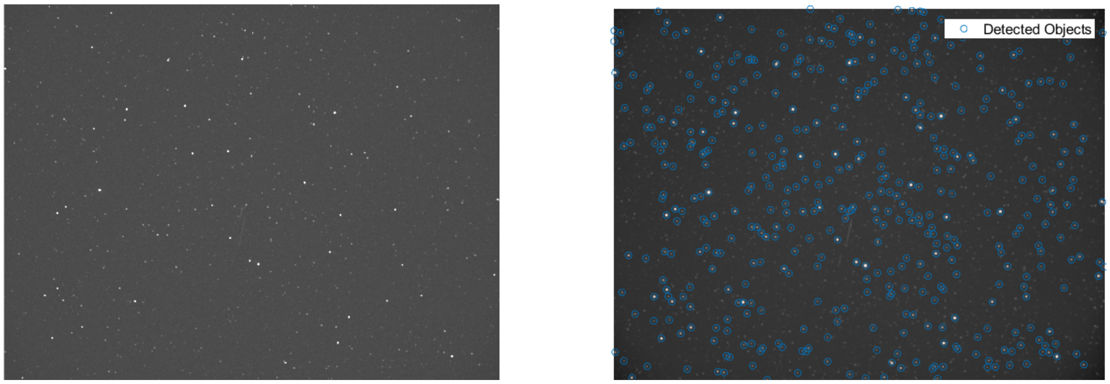
However, if we take the same raw image, and apply the image calibration and noise reduction processes of sections 3.2 and 3.2.6, nearly all the stars visible in the image are detected as objects as shown in Figure 4.2.



(a) Raw Image

(b) Processed Image

Figure 4.1. Objects Detected, No Calibration, No Noise Reduction

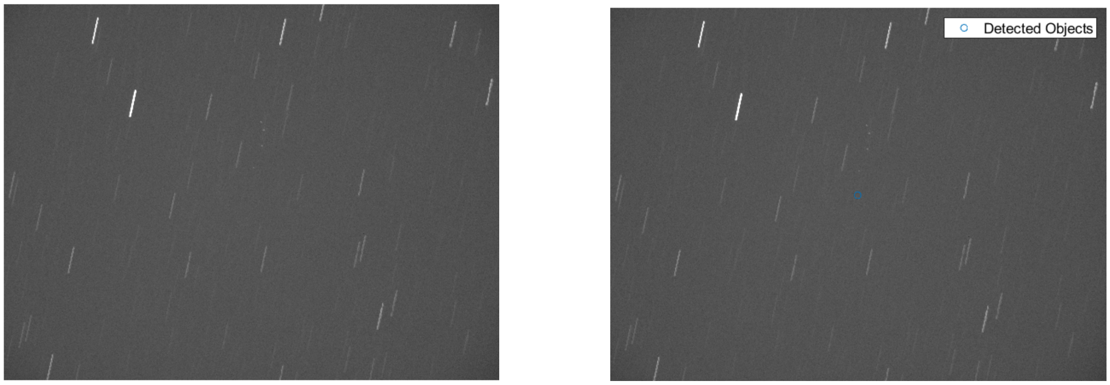


(a) Raw Image

(b) Processed Image

Figure 4.2. Objects Detected, Calibration + Noise Reduction Applied

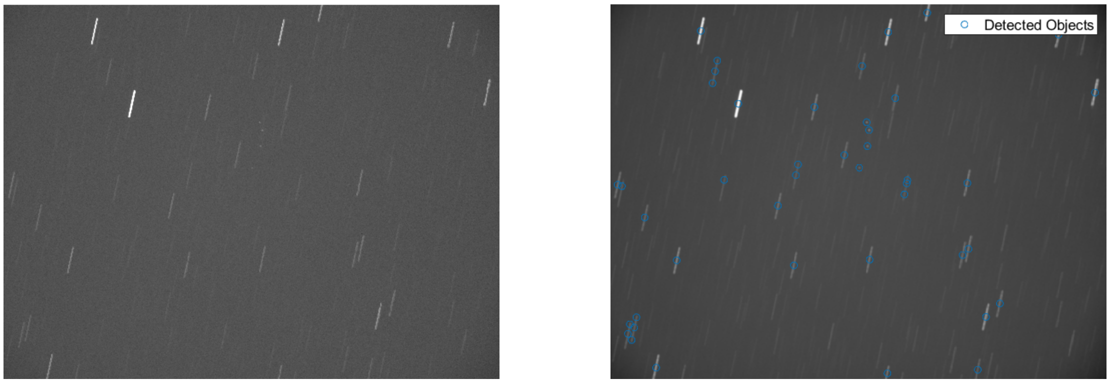
The same results are seen with images taken in RSO tracking mode. When just the raw image is used to attempt centroiding and object detection, there is not enough contrast between the background noise of the image to separate individual objects in the image. Figure 4.3 shows the results of applying no processing for the centroiding process. Following the same result as above, if we apply image calibration and noise reduction, the results are greatly improved and most of the objects visible are detected. Figure 4.4 shows the results of applying noise reduction and image calibration to the same raw image as Figure 4.1(a).



(a) Raw Image

(b) Processed Image

Figure 4.3. RSO Tracking Mode, No Calibration, No Noise Reduction Results



(a) Raw Image

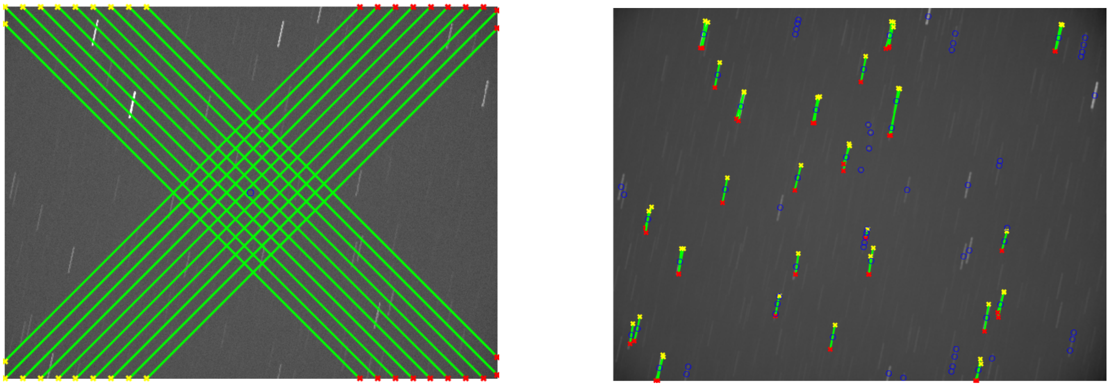
(b) Processed Image

Figure 4.4. RSO Tracking Mode, Calibration + Noise Reduction Results

For images taken with either tracking mode, adjusting the threshold filter cutoff limit will allow the detection of fainter objects. However, care must be taken when choosing a threshold level. Setting the threshold level too low results in the background noise not being sufficiently reduced to allow successful object detection.

4.1.2 Line Detection Results

Detecting objects successfully is important, however, without knowing what type of object has been detected, nothing further can be done. To proceed, all objects must then be labeled as point-source objects or lines in order to determine whether an object is a star or candidate RSO. The result of applying the Hough Transform for line detection to a raw image and then a calibrated and noise reduced image of INTELSAT30 are shown in Figure 4.5. In the case where no image calibration or noise reduction was applied, individual objects were not detected in the image and the only lines detected in the image spanned the entire image. (Note that the Hough Transform has some settings that can be adjusted to change the line detection thresholds.) Applying noise reduction and image calibration led to successful line identification for many of the lines in the image.



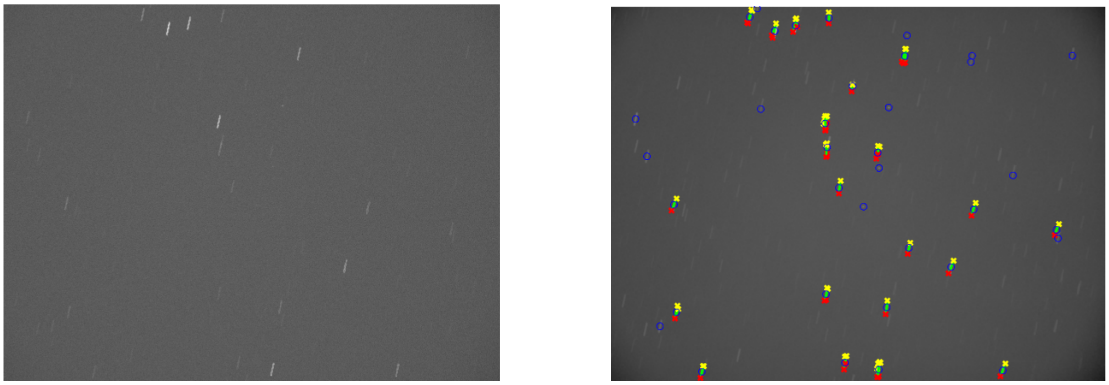
(a) Raw Image Lines Detected

(b) Processed Image Lines Detected

Figure 4.5. RSO Tracking Mode Results: Raw and Processed Image Hough Line Results

The results of applying noise reduction and image calibration before the line detection step for an image of MEXSAT3 are shown in Figure 4.6. Note that fewer lines are detected in this image due to the lower overall contrast. As before, without applying noise reduction and image calibration, no objects are detected (not shown).

While detecting objects and lines is important, the end goal here is to be able to discriminate between objects in order to discover which objects represent candidate RSOs. Once detected in a series of images, candidate RSO locations are used to find an orbit estimate using Gooding’s Method (section 2.7.2).



(a) Raw Image

(b) Processed Image Lines Detected

Figure 4.6. RSO Tracking Mode Results: MEXSAT3 Raw and Hough Line Results

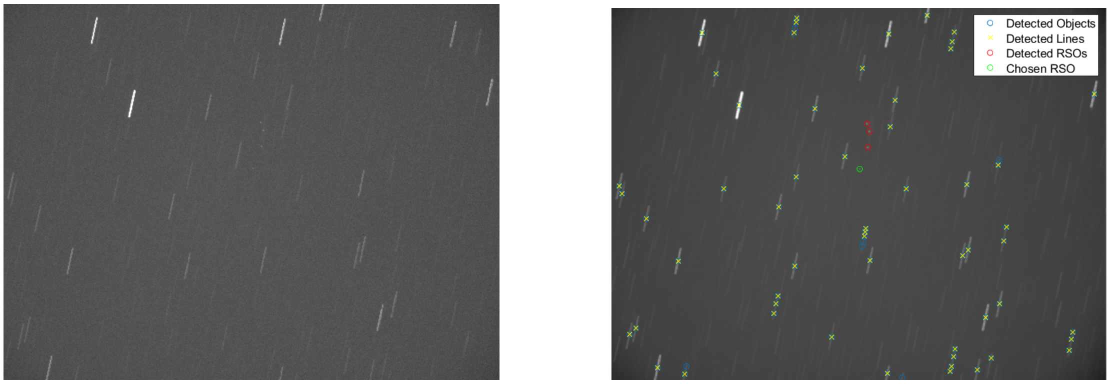
4.1.3 RSO Detection Results

The RSO detection and discrimination algorithm presented in section 3.4.1 was applied to various images captured of both INTELSAT30 and MEXSAT3. In most images, the algorithm was successfully able to identify all RSOs in a given image.

INTELSAT30

All 69 images taken in RSO tracking mode of INTELSAT30 were tested using the RSO identification algorithm from section 3.4.2. All images were calibrated and had noise reduction applied. Figure 4.7 shows the successful identification of four RSOs in an image of INTELSAT30. Also, nearly all streaks corresponding to stars were correctly identified. This image was a 20 second exposure captured in RSO tracking mode.

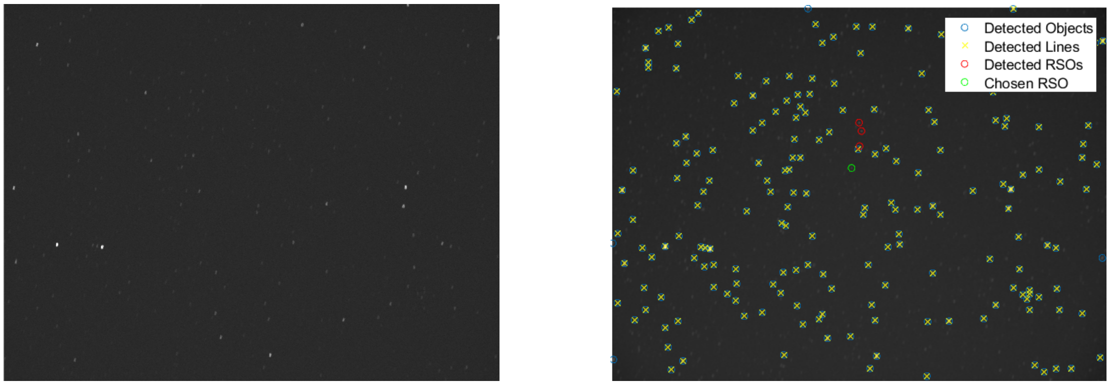
Figure 4.7 shows a case where by inspection it is fairly obvious which objects in the image correspond to stars, and which to RSOs. Given an image with a much shorter exposure time, this distinction is not easy to make. Figure 4.8 shows an RSO tracking mode image of INTELSAT30 with an exposure time of 2 seconds. The stars streak, but



(a) Raw Image (b) Processed Image

Figure 4.7. INTELSAT30 Successful RSO Detection Results

only a a few pixels as shown in the zoomed view of Figure 4.9. All four RSOs present in the image were correctly identified as well as many of the stars in the image.



(a) Raw Image (b) Processed Image

Figure 4.8. INTELSAT30 Successful RSO Detection Results: 2 Second Exposure

MEXSAT3

The images captured of MEXSAT3 are similar in many respects to those captured of INTELSAT30. There is quite a bit of contrast variation in the images of MEXSAT3. Most images were captured using a 10 second exposure, which seemed to give a balance between capturing enough light from the RSOs and letting the stars streak long enough to help with line identification. Figure 4.10 shows a 10 second image taken in RSO tracking

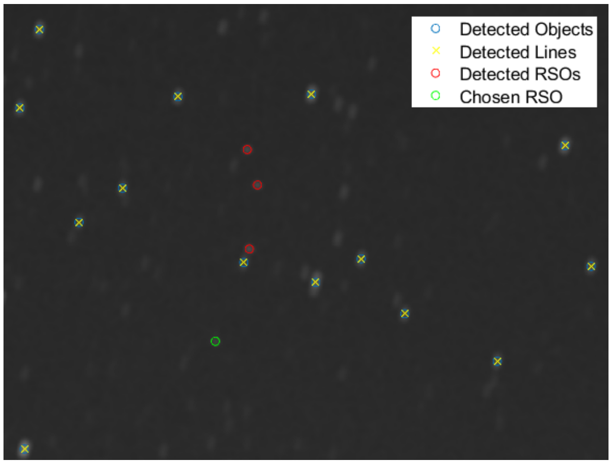
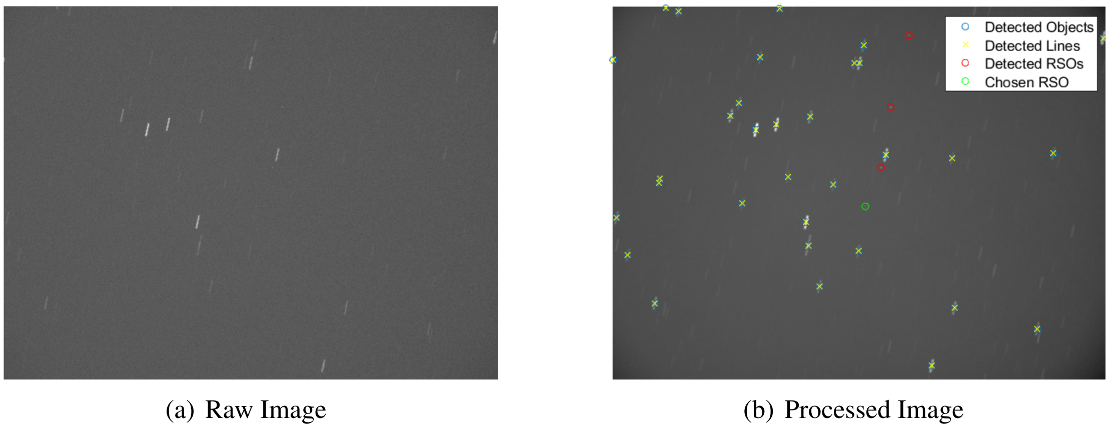


Figure 4.9. Center Crop, Successful RSO Detection from 2 Second Exposure mode of MEXSAT3. The raw image is displayed side by side with the final processed image. Once again all RSOs present in the image were found and labeled correctly.



(a) Raw Image

(b) Processed Image

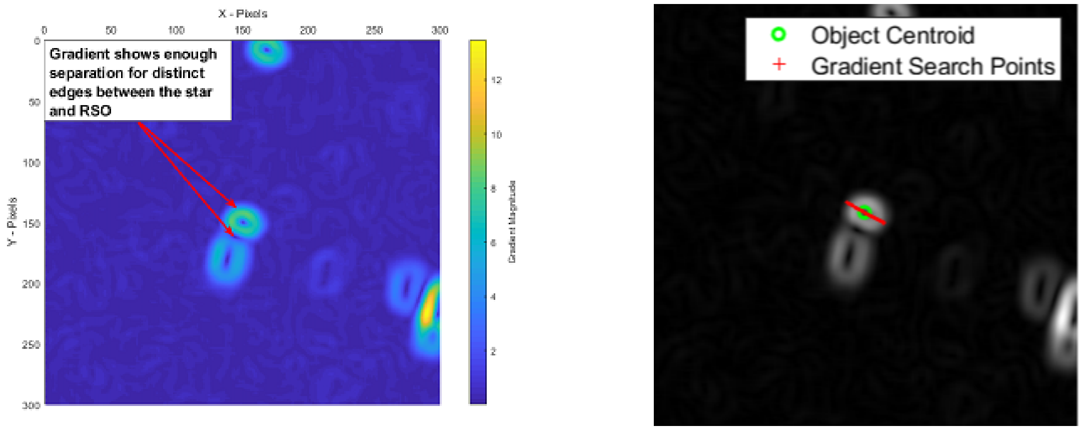
Figure 4.10. MEXSAT3 Successful RSO ID: 10 Second Exposure

Out of 44 images of MEXSAT3, all RSOs were found in 39 of the images. The five images that did not successfully find all RSOs did in some cases succeed in finding some of the RSOs. The others represent difficult cases which are discussed in the next section. From the results thus far, RSO detection results have been made possible in the experimental images only with the application of at least some noise reduction and image calibration.

Difficult Case Results

There are certain images captured that represent “difficult” cases. Some of the difficulties come from a very short exposure time while in RSO tracking mode, resulting in all the stars barely streaking through the image. Another difficult case comes from the image set of MEXSAT3, when for the last few frames, the Orion Nebula enters the frame and throws off the image contrast.

One difficult case that occurs multiple times in the data sets is the case when an RSO overlaps a star in the image. Figure 4.11 shows one such case where the point-source RSO signal overlaps with a short streak of a star. In this case there was enough of a difference in the gradients of each object for separate objects to be detected and the RSO was successfully identified.

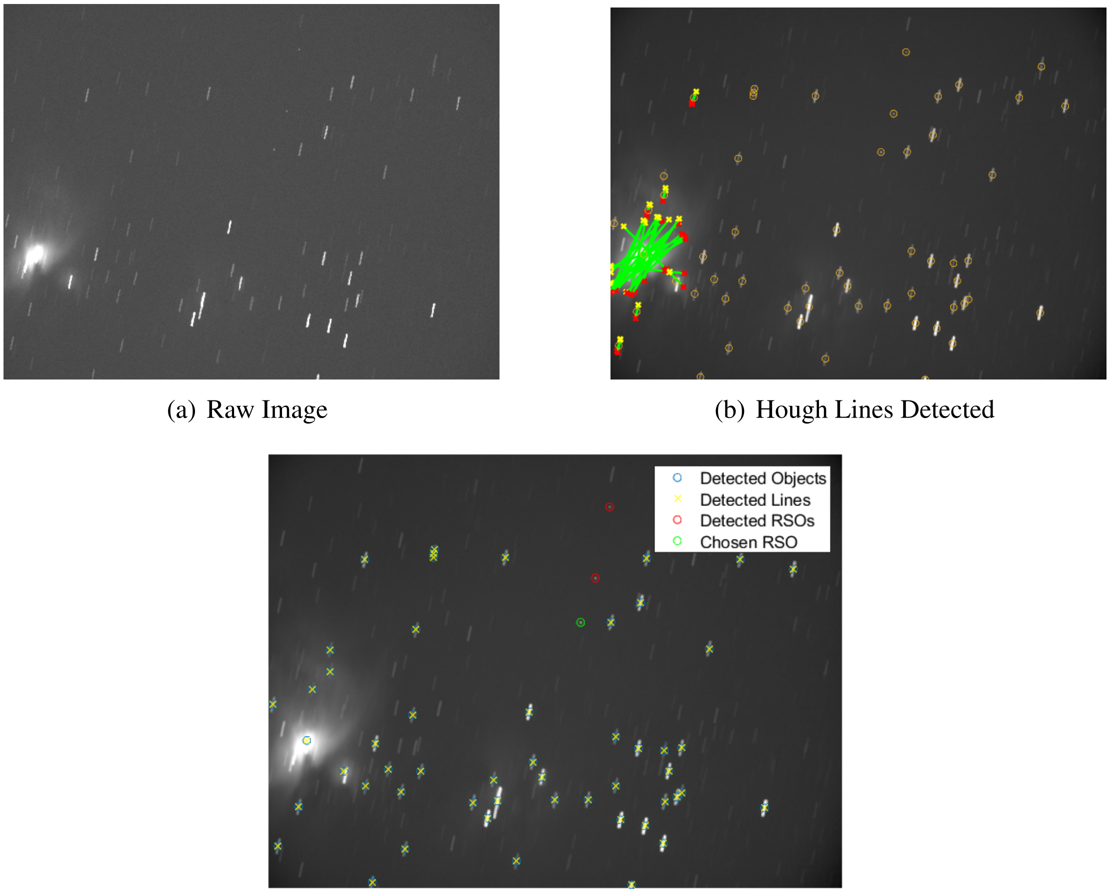


(a) Raw Image

(b) Processed Image

Figure 4.11. INTELSAT30 Overlapping Star

Figure 4.12(a) shows a montage of one of the most difficult images for object identification that was obtained. The target satellite, MEXSAT3, passed through the same field of view as the Orion Nebula, which can be seen in the lower left of the image. However, the RSO identification algorithm correctly identified the three RSOs present in the image and managed to label most of the stars as streaks as seen in Figure 4.12(c). Figure 4.12(b) shows a disadvantage of using a Hough Transform as the only method of line detection. In images with large non-stellar objects, line segments can be found that do not correspond to star or RSO streaks. However, the gradient-search method of object identification was successfully able to label the majority of the stars as lines in the image.



(a) Raw Image

(b) Hough Lines Detected

(c) Successful RSO Identification Results

Figure 4.12. MEXSAT3 Successful RSO ID: Orion Nebula Enters Image Frame

Figure 4.13 shows the distribution of detected RSOs for the INTELSAT30 image set (note, some images were removed from consideration as they were of very poor quality). The number of RSOs that should be detected in an given image should be four, a mark that was attained by 44 out of 59 images. The remaining images were from difficult cases like those discussed in section 4.1.3.

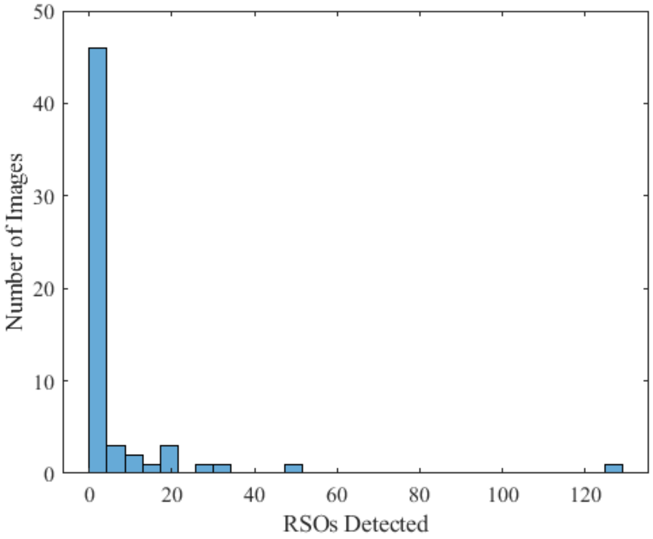


Figure 4.13. INTELSAT30: Overall RSO Detection Results

The overall effectiveness of the RSO detection process can be examined by the histogram plot shown in Figure 4.14. The number of RSOs that should be detected in an image of the MEXSAT data set is 3 or 4. The MEXSAT3 grouping of satellites was more spread out than the INTELSAT30 group so in some frames only 3 of the 4 satellites are visible. For the images of MEXSAT3, 2 out of 44 images registered false positives, where a false positive comes from an image with a detected number > 4. The four images that detected only two RSOs were some of the difficult cases discussed above where not all the objects were successfully identified.

4.1.4 Gradient Search Method Failures

There were times that the gradient search method failed to identify an object. There are two tuning parameters in the gradient search, for how many steps up, and how many steps down in a row should be considered for an edge. Most failures of the gradient search

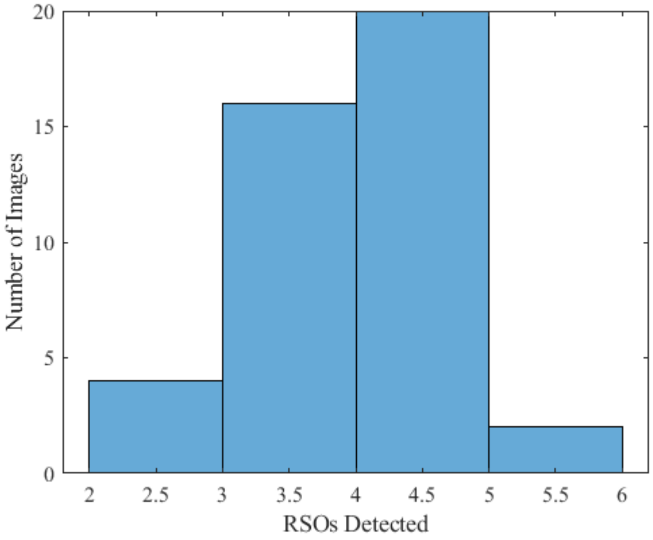


Figure 4.14. MEXSAT3: Overall RSO Detection Results

method occurred because these thresholds were not met for a few objects. In cases that do not detect objects, these objects are labeled as unknown.

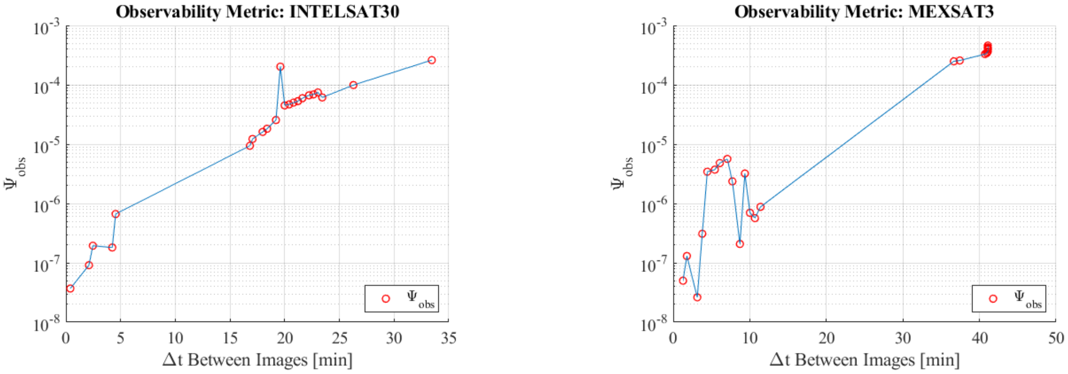
4.2 Observability Metric Results

Now that RSOs can be successfully identified in most arbitrary images that were captured for this research, the next task is to run the detected RSOs through the orbit determination process. We will also make use of the Observability Metric Ψ_{obs} developed in section 3.9. Theoretically as the spacing increases between images, the value of Ψ_{obs} should increase and the resulting orbit estimate should improve.

4.2.1 Experimental Results

In order to examine what Ψ_{obs} tells us about the resulting orbit solution, the sets of images from Test Case A and Test Case B (INTELSAT30 and MEXSAT3 respectively) were processed and all frames with successful RSO discrimination were used to form different combinations of LOS vectors. The average time between observations was varied before running the resulting measurements through Gooding to get an orbit estimate.

Figure 4.15 shows the results of increasing the average time between images and the resulting value of Ψ_{obs}.



(a) INTELSAT30

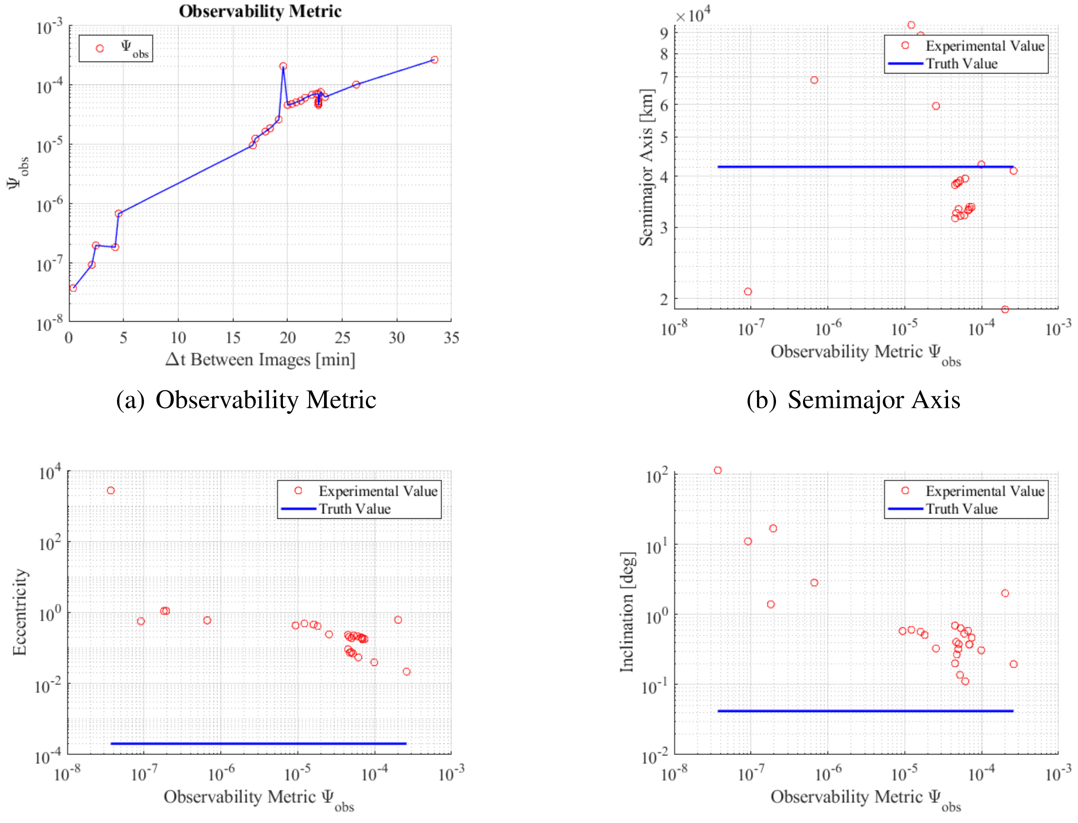
(b) MEXSAT3

Figure 4.15. Observability Metric

Now some estimate of the quality of the orbit estimate needs to be made. Therefore the orbit results are plotted against the known truth model obtained from a TLE file nearest the epoch of the observations. The first three orbital elements are chosen as the comparison point as they help describe the orbit shape.

Figure 4.16 shows the results of running different sets of measurements obtained from INTELSAT30 through the AIOD process. Figure 4.16(a) shows the upward trend that Ψ_{obs} follows as the time between images is increased. Logically this makes sense, as we increase Δt between images, then we are observing more of a satellite's orbit, which should give a better orbit solution. Figure 4.17 shows similar results for the image set obtained from MEXSAT3. In this case as well we see a general trend of the orbital elements moving closer to the truth value (shown by the solid blue line) as Ψ_{obs} increases.

If our hypothesis is correct, the higher the value of Ψ_{obs} becomes, the more the orbit solution should improve. The following series of plots show the resulting errors



(a) Observability Metric

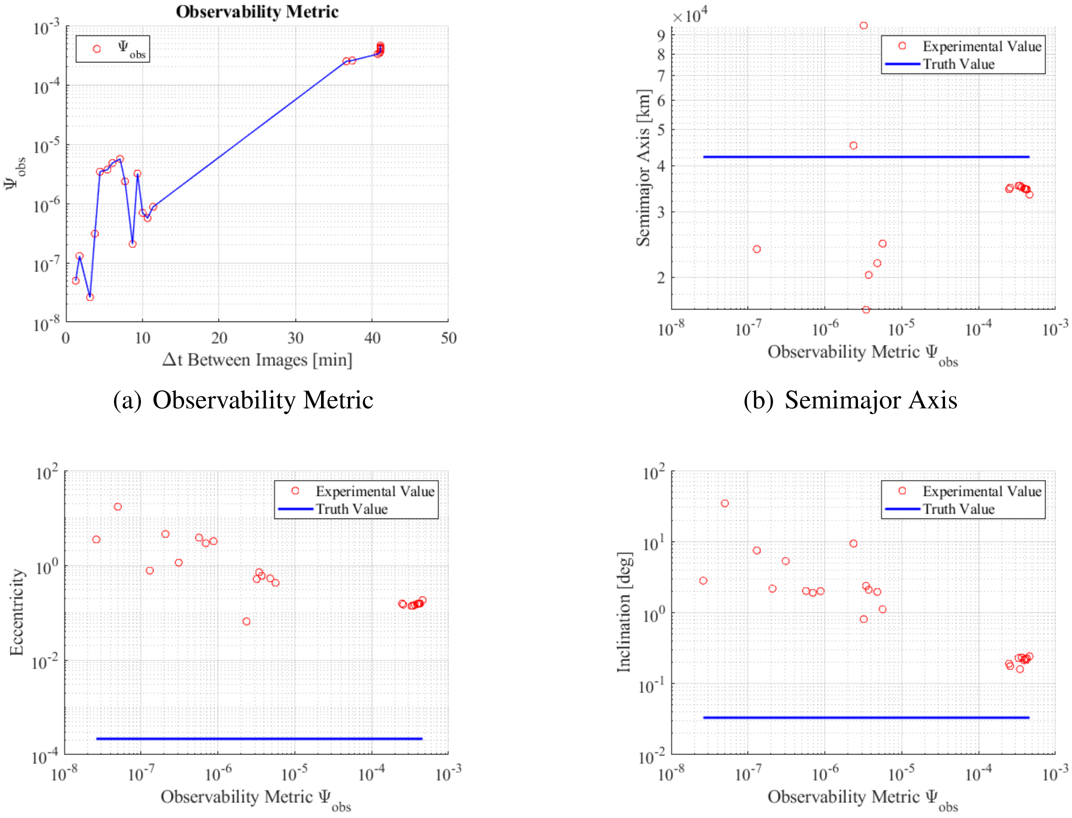
(b) Semimajor Axis

(c) Eccentricity

(d) Inclination

Figure 4.16. Observability Metric Results: INTELSAT30

in the first three orbital elements. Figure 4.18 shows what we might expect, a general decreasing trend in the error as the value of Ψ_{obs} increases. Again, logically, this makes sense. As we seek to increase Ψ_{obs}, we are trying to use the most linearly independent measurements, which should lead to a better orbit estimate. From Figures 4.16(c) and 4.16(d), we notice that the error in eccentricity and inclination appears large. However, it must be remembered that the true values of eccentricity and inclination are both near zero, while the estimated values are on the order of 1×10^{-2}. Thus, although the error amount seems large this is a byproduct of the actual number being so diminutive.



(a) Observability Metric

(b) Semimajor Axis

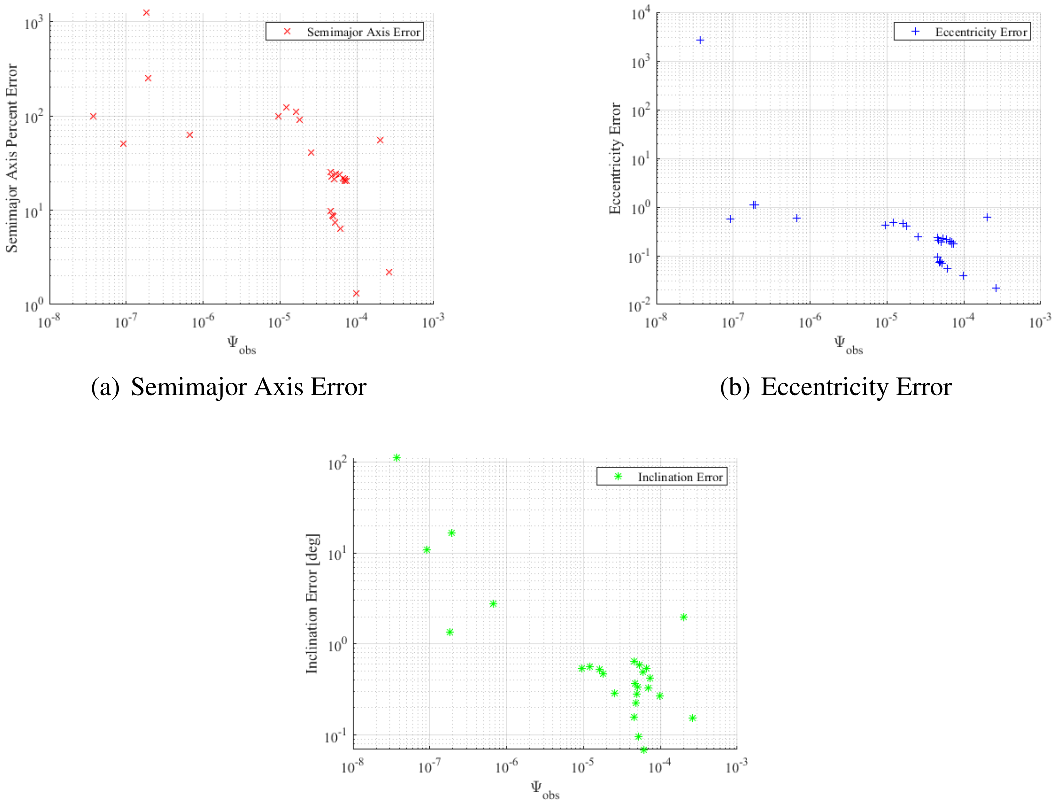
(c) Eccentricity

(d) Inclination

Figure 4.17. Observability Metric Results: MEXSAT3

4.2.2 Simulation Results

The first scenario considered is a ground-based observer targeting the GEO satellite INTELSAT30 (defined as Test Case A in Table 4.1). The regular Gooding's method with three measurements was used to generate the orbit for various image spacing amounts between simulated measurements. Uniform spacing between observations is assumed for the simulated measurements. Simulated observations were created using the same location, time, and image spacing as the experimental measurements to provide a direct comparison between theory and practice. Figure 4.20 shows the characteristic trend in Ψ_{obs} as the time between images increases. Note that this simulation assumes zero noise.



(a) Semimajor Axis Error

(b) Eccentricity Error

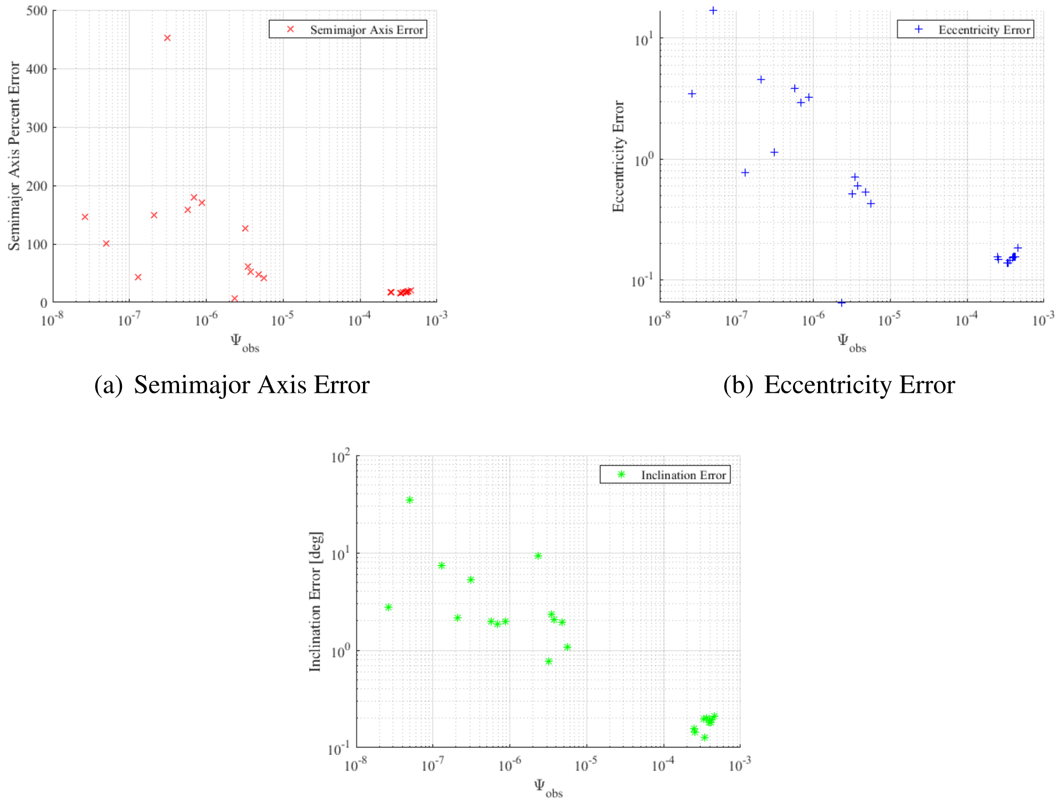
(c) Inclination Error

Figure 4.18. INTELSAT30 Orbital Elements Error and Observability Metric

The result agrees with what was found for the experimental image measurements from section 4.2.1.

Monte-Carlo Noise Simulation

In order to examine the effects of noise on the Ψ_{obs} and the resulting orbit estimate, Monte-Carlo trials were run to generate observations of INTELSAT30 for the specific measurement spacing values of $\Delta t = 3.83, 10$, and 25 minutes. In order to provide a comparison point, the same site location and epoch were used as the experimental measurements. Seven simulations were run with increasing noise values.



(a) Semimajor Axis Error

(b) Eccentricity Error

(c) Inclination Error

Figure 4.19. MEXSAT3 Orbital Elements Error and Observability Metric

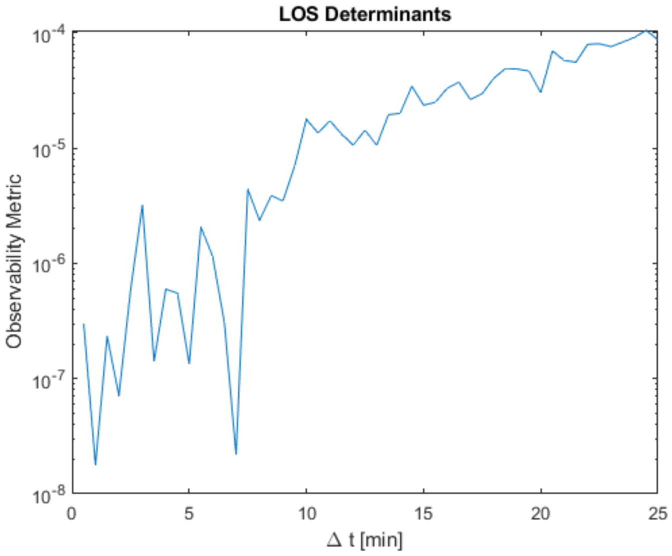


Figure 4.20. Observability Metric with Increasing Time Between Measurements

Table 4.2

Simulation Results: 0.1 - Arcsecond Noise Added

Δt [min]	3.83	10	15	25
Ψ_{obs}	3.92E-07	6.96E-06	2.34E-05	1.08E-04
Semimajor Axis (a) [km]	42223	42165	42163	42164
1-σ [km]	1.34E+03	1.93E+02	8.58E+01	3.10E+01
error [km]	5.69E+01	1.28E+00	2.38E+00	2.20E+00
Eccentricity (e)	1.81E-02	2.67E-03	1.21E-03	4.94E-04
1-σ	1.51E-02	2.18E-03	9.54E-04	3.20E-04
error	1.79E-02	2.46E-03	1.01E-03	2.89E-04
Inclination (i) [deg]	5.04E-02	4.17E-02	4.16E-02	4.15E-02
1-σ [deg]	2.60E-02	4.49E-03	1.98E-03	6.94E-04
error [deg]	8.53E-03	2.27E-04	3.44E-04	3.63E-04

Table 4.2 shows the simulation results with the lowest noise level simulated. Tables A.1 - A.6, which can be found in Appendix A, show the results of adding different levels of Gaussian noise to the simulated measurements of INTELSAT30. The first three orbital elements (semimajor axis (a), eccentricity (e), and inclination (i)) are listed along with their respective $1-\sigma$ standard deviations and error amounts. The overall trend resulting from all noise levels shows that with increasing Δt, the value of Ψ_{obs} increases and the orbit solution improves. Again, this makes sense. The longer the time between measurements, the more of the orbit arc is observed, which results in a better orbit estimate. As noise is added to the measurements, the mean orbital element values converge towards the true values, however, with each increase in noise, the standard deviation on those mean values increases greatly. Take for example, the case of $\Delta t = 25$, the 1-σ for the semimajor axis found with a noise level of 0.1 arcesconds is 31 *km*, but that for the noise level of 10 arcseconds is 3.3×10^3 *km*.

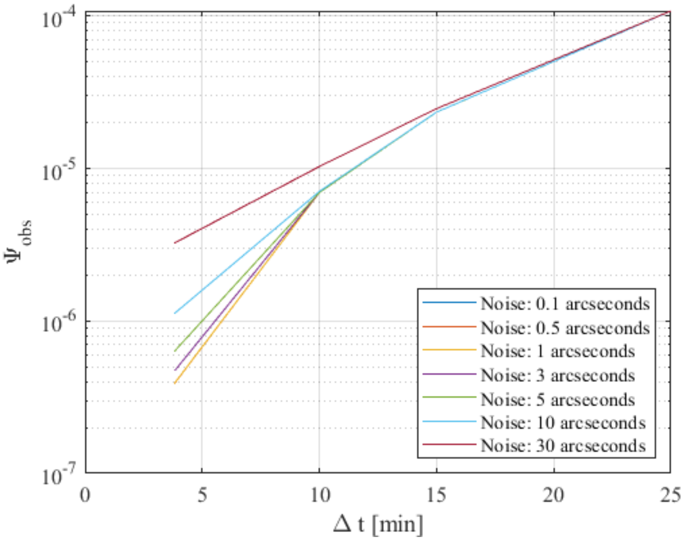


Figure 4.21. Simulation Results: Observability Metric and Noise

Another interesting result from the simulation results can be seen in Figure 4.21. At higher noise values, the value of Ψ_{obs} increases more for lower measurement times. In this case, if we imagine our simulated measurement as a unit vector in 3D space, when we add noise, that unit vector is allowed to wobble in 3D space creating a cone. So physically it makes sense that Ψ_{obs} would increase with more noise because the cone subtended by the measurements will increase as more noise is added, making the measurements in a sense, less linearly independent. However, for these instances, that will not give a better orbit estimate just because the value of Ψ_{obs} was higher. In fact Tables A.1 - A.6 show that for the cases of high noise and low Δt, the orbit estimate is much worse. The increase in the value of Ψ_{obs} is a by product of using a Gaussian distribution for the noise to generate the measurements, and may not be the case given real measurement noise. Thus for the case of very noisy images, the metric Ψ_{obs} could lead to choosing noisier measurements.

5. Conclusion

5.1 Contributions

The gradient based search algorithm for RSO identification successfully identified RSOs in various experimental images. Several cases with less than ideal images were shown to still have RSOs correctly discriminated. Noise reduction techniques proved important to allowing successful operation of the new method. A key result comes from the fact that only one image is needed for RSO detection with this method. Given a sparse data set, with images spread over a large arc, the only option for RSO identification is from single images.

A new metric for gauging the quality of observations for the AIOD process was empirically demonstrated. The orbit was shown to improve with increasing values of the metric Ψ_{obs}. The metric was tested against both simulated measurements and various experimental measurements. In the future, this metric could provide a way for increasing the autonomy of the AIOD process.

The end-to-end angles-only orbit determination process using small aperture optical systems was successfully demonstrated. Coupled with the image processing algorithms and RSO identification routine, orbit estimates of two GEO satellites were computed from imagery obtained from an 11 inch ground based telescope.

5.2 Future Work

Future work recommendations fall into two main categories, advancing the autonomy of the AIOD process and adding improved capabilities for object detection.

5.2.1 RSO Identification from Gross Motion

Having the ability to identify RSOs from a single image was successfully demonstrated. Single frame RSO ID can be improved further by examining the gross motion of all objects in a given image. Taken in RSO tracking mode, the stars of an image all form streaks with a common orientation. Sometimes, while tracking one set of RSOs, another will pass through the frame and streak through on a different trajectory than the stars. An example of such a serendipitous detection is shown in Figure 5.1.

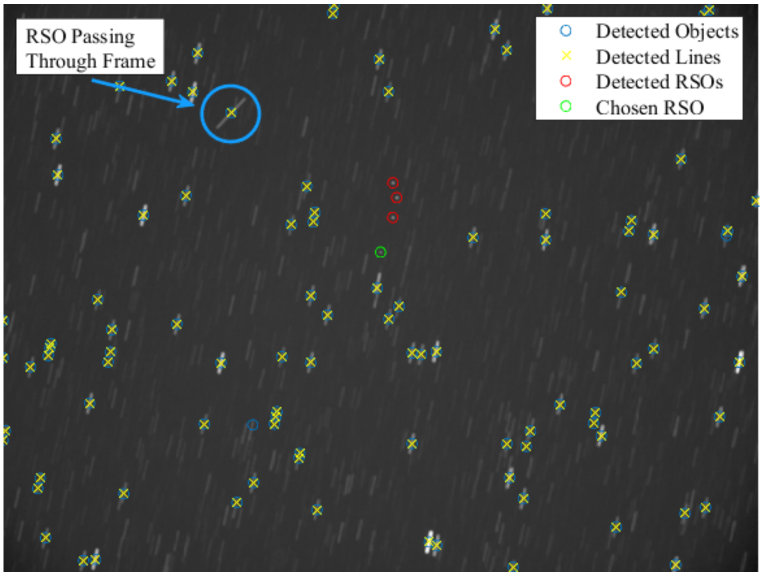


Figure 5.1. RSO Passing Through Frame

The RSO passing through at a different orientation was successfully identified as a line object by the existing RSO detection algorithm. However, it was not labeled as an RSO. Taking into account the different orientation of the line from the majority of line objects (stars) could allow detection of such objects.

5.2.2 Tracking GEO Constellations via Template Matching

Template matching holds some promise to application in the identification of GEO constellations from frame to frame. Once a GEO constellation has been successfully identified, the relative position between the satellites remains constant across any number of images. Running a template match formed from the first positive identification could allow for a quick way to correctly associate GEO objects across any number of frames.

5.2.3 Extended Objects

Extended objects pose a unique problem for orbit estimation because of their resolved nature. Further research is needed in how to choose a point for running the AIOD process from a series of images for extended objects. This has application for both AIOD and proximity operations.

5.2.4 Autonomous Surveillance

The techniques demonstrated in this research have been developed with the future goal of developing an autonomous AIOD system. Folding in the observability metric allows for a system to choose among the best images from any data set. Automatic detection has already been demonstrated (section 4.1). Coupling automatic object detection with an onboard catalog of objects is the next step. The end goal would be to produce a unified framework where a telescope could find an object, begin tracking and imaging to build an onboard catalog. Once enough images are taken, then the AIOD process would be performed to generate an orbit estimate. Propagating this estimate forward to locate the same object could then be used to gain more observations and further improve the orbit estimate.

5.2.5 N-Measurements

Cases where N-measurements were used for the AIOD process and the observability metric Ψ_{obs} should be examined. A full extension of Ψ_{obs} for $N > 3$ measurements should be performed in order to allow autonomous selection of data for the AIOD process to use in N-measurement algorithms and filtering applications.

5.3 Concluding Remarks

Creating a system of small aperture optical telescopes for space surveillance has nearly all the necessary pieces to become reality. The conclusions of this research can be used to provide tech-savvy amateur astronomers with the tools necessary to provide a vast store of much needed space surveillance data. As was shown by this research, all that is needed to provide SSA data is a telescope capable of tracking satellites and a camera.

REFERENCES

Ascom - standards for astronomy. (n.d.). Retrieved from https://ascom-standards.org/

Astrometry.net. (n.d.). Retrieved from http://astrometry.net/

Astrometry.net code readme. (n.d.). Retrieved from https://astrometrynet.readthedocs.io/en/latest/readme.html

Astrophotography tool. (n.d.). Retrieved from https://ideiki.com/astro/Default.aspx

Ballard, D. H., & Brown, C. M. (1982). *Computer vision.* Englewood Cliffs, NJ: Prentice Hall.

Battin, R. H. (1999). *An introduction to the mathematics and methods of astrodynamics.* Reston, VA: American Institute of Aeronautics and Astronautics, Inc.

Berry, R., & Celestron Engineering Team. (2016, Apr). Big! fast! wide! sharp! the story of the rowe-ackermann schmidt astrograph. Retrieved from https://celestron.comhttps://s3.amazonaws.com/celestron-site-support-files/support_files/rasa_white_paper_web.pdf

Branham, R. (2005, January). Laplacian orbit determination and differential corrections. *Celestial Mechanics and Dynamical Astronomy*, *93*, 53-68. doi: 10.1007/s10569-005-3242-6

Brogan, W. L. (1985). *Modern control theory.* Englewood Cliffs, NJ: Prentice-Hall.

Crassidis, J., & Markley, L. (2003, 08). Unscented filtering for spacecraft attitude estimation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit* (Vol. 26, p. 536-542). doi: 10.2514/6.2003-5484

DARPA. (n.d.). Orbit outlook. Retrieved from https://www.darpa.mil/program/orbitoutlook

Eccles, J. (2015). Satellite tracker. Retrieved from https://www.heavenscape.com/

Escobal, P. R. (1976). *Methods of orbit determination.* Huntington, NY: R. E. Krieger Pub. Co.

Galasso, A. (n.d.). ansvr - local astrometry.net plate solver for windows. Retrieved from https://adgsoftware.com/ansvr/

G. Harris, C., & J. Stephens, M. (1988, 01). A combined corner and edge detector. *Proceedings 4th Alvey Vision Conference*, *1988*, 147-151. doi: 10.5244/C.2.23

Gooding, R. H. (1990, 06). A procedure for the solution of lambert's orbital boundary-value problem. *Celestial Mechanics and Dynamical Astronomy*, *48*, 145-165. doi: 10.1007/BF00049511

Gooding, R. H. (1993). *A new procedure for orbit determination based on three lines of sight (angles only)*. United Kingdom: Defence Research Agency Farnborough.

Gooding, R. H. (1996, Dec 01). A new procedure for the solution of the classical problem of minimal orbit determination from three lines of sight. *Celestial Mechanics and Dynamical Astronomy*, *66*(4), 387–423. doi: 10.1007/BF00049379

Henderson, T. A., Mortari, D., & Davis, J. (2010). Modifications to the gooding algorithm for angles-only initial orbit determination. In *AAS/AIAA Space Flight Mechanics Meeting, San Diego.*

Image gradient. (2018, Sep). *Wikipedia*. Retrieved from `https://en.wikipedia.org/wiki/Image_gradient`

Izzo, D. (2015, May). Revisiting lambert's problem. *Celestial Mechanics and Dynamical Astronomy*, *121*(1). doi: 10.1007/s10569-014-9587-y

Julier, S., Uhlmann, J., & F. Durrant-Whyte, H. (2000, 04). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, *45*, 477 - 482. doi: 10.1109/9.847726

Karimi, R. R., & Mortari, D. (2014, 01). A performance based comparison of angle-only initial orbit determination methods. In *AAS/AIAA Advanced Astronautical Sciences, Hilton Head Island, SC* (Vol. 150, p. 1793-1809).

Kim, S.-G., Crassidis, J., Cheng, Y., M. Fosbury, A., & Junkins, J. (2007, 01). Kalman filtering for relative spacecraft attitude and position estimation. *Journal of Guidance Control and Dynamics*, *30*. doi: 10.2514/1.22377

Lang, D., Hogg, D. W., Mierle, K., Blanton, M., & Roweis, S. (2010). Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, *139*(5), 17821800. doi: 10.1088/0004-6256/139/5/1782

Lewis, F. L., Syrmos, V. L., & Vrabie, D. L. (2012). *Optimal control*. Hoboken, NJ: Wiley.

Martínez, F., Águeda Mat, A., Grau, J., Fernndez Snchez, J., & Aivar Garca, L. (2012, 01). Comparison of angles only initial orbit determination algorithms for space debris cataloguing. *Journal of Aerospace Engineering, Sciences and Applications*, *4*. doi: 10.7446/jaesa.0401.04

Montojo, F., López Moratalla, T., & Abad, C. (2011, 03). Astrometric positioning and orbit determination of geostationary satellites. *Advances in Space Research*, *47*, 1043-1053. doi: 10.1016/j.asr.2010.11.025

Mortari, D., Bruccoleri, C., La Rosa, S., & Junkins, J. (2002). CCD data processing improvements. In *Proceedings of the International Conference on Dynamics and Control of Systems and Structures in Space, King College, Cambridge, England.*

Newton's laws of motion: Glenn research center. (2019). Retrieved from https://www.grc.nasa.gov/www/k-12/airplane/newton.html

On, J. (2016, Dec). Read noise in CCD cameras. Retrieved from https://www.atik-cameras.com/news/read-noise-ccd-cameras/

Orbital elements. (n.d.). Retrieved from https://spaceflight.nasa.gov/realdata/elements/graphs.html

Prussing, J. E., & Conway, B. A. (1993). *Orbital mechanics*. New York: Oxford University Press.

Schaeperkoetter, A. (2011, 01). A comprehensive comparison between angle-only initial orbit determination techniques (Master's thesis). Retrieved from http://hdl.handle.net/1969.1/ETD-TAMU-2011-12-10242

Schaub, H., & Junkins, J. L. (2003). *Analytical mechanics of space systems*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc.

Sease, B., Flewelling, B., & Black, J. (2017, 02). Automatic streak endpoint localization from the cornerness metric. *Acta Astronautica*, *134*, 345-354. doi: 10.1016/j.actaastro.2017.02.009

Shi, J., & Tomasi, C. (1994, June). Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA* (p. 593-600). doi: 10.1109/CVPR.1994.323794

Stellarium astronomy software. (n.d.). Retrieved from https://stellarium.org/

Strang, G. (2003). *Introduction to linear algebra*. Wellesley, MA: Cambridge Press.

Teledyne Photometrics. (n.d.). Dark current. Retrieved from https://www.photometrics.com/resources/learningzone/darkcurrent.php

Tiwari, M., Zuehlke, D., & Henderson, T. (2019, 01). Identifying spacecraft configuration using deep neural networks for precise orbit estimation. In *AAS/AIAA Spaceflight Mechanics Meeting, Ka'anapali, HI.*

Vallado, D. A. (2010, 01). Evaluating gooding angles-only orbit determination of space based space surveillance measurements. In *AAS Born Symposium.*

Vallado, D. A., & McClain, W. D. (1997). *Fundamentals of astrodynamics and applications*. New York: McGraw-Hill.

Wan, E., & Van Der Merwe, R. (2000, 02). The unscented kalman filter for nonlinear estimation. In *IEEE 2000, Adaptive Systems for Signal Processing, Communications, and Control Symposium, Lake Louise, Alberta Canada, Canada* (p. 153 - 158). doi: 10.1109/ASSPCC.2000.882463

ZWO Company. (2016, May). ASI1600MM pro (mono). Retrieved from `https://astronomy-imaging-camera.com/product/asi1600mm-cool`

A. Monte-Carlo Simulation Results

Table A.1

Simulation Results: 0.5 Arcsecond Noise Added

Δt [min]	3.83	10	15	25
Ψ_{obs}	3.92E-07	6.96E-06	2.34E-05	1.08E-04
Semimajor Axis (a) [km]	45178	42195	42170	42164
1-σ [km]	2.41E+05	9.74E+02	4.32E+02	1.55E+02
error [km]	3.01E+03	2.94E+01	4.65E+00	1.71E+00
eccentricity (e)	9.56E-02	1.33E-02	5.93E-03	2.15E-03
1-σ	9.83E-02	1.10E-02	4.89E-03	1.75E-03
e error	9.54E-02	1.31E-02	5.73E-03	1.95E-03
Inclination (i) [deg]	1.59E-01	4.64E-02	4.24E-02	4.16E-02
1-σ [deg]	1.30E-01	1.95E-02	9.47E-03	3.44E-03
error [deg]	1.17E-01	4.49E-03	4.99E-04	2.59E-04

Table A.2

Simulation Results: 1 - Arcsecond Noise Added

Δt [min]	3.83	10	15	25
Ψ_{obs}	3.92E-07	6.96E-06	2.34E-05	1.08E-04
Semimajor Axis (a) [km]	50523	42316	42191	42167
1-σ [km]	1.96E+06	1.99E+03	8.70E+02	3.12E+02
error [km]	8.36E+03	1.50E+02	2.56E+01	1.13E+00
Eccentricity (e)	2.54E-01	2.67E-02	1.19E-02	4.29E-03
1-σ	7.34E-01	2.24E-02	9.89E-03	3.53E-03
e error	2.54E-01	2.65E-02	1.17E-02	4.09E-03
Inclination (i) [deg]	3.55E-01	6.07E-02	4.56E-02	4.20E-02
1-σ [deg]	6.09E-01	3.65E-02	1.75E-02	6.78E-03
error [deg]	3.13E-01	1.88E-02	3.66E-03	1.16E-04

Table A.3

Simulation Results: 3 - Arcsecond Noise Added

Δt [min]	3.83	10	15	25
Ψ_{obs}	4.72E-07	6.96E-06	2.34E-05	1.08E-04
Semimajor Axis (a) [km]	41153	43785	42430	42202
1-σ [km]	2.78E+06	7.30E+04	2.73E+03	9.40E+02
error [km]	1.01E+03	1.62E+03	2.64E+02	3.64E+01
Eccentricity (e)	2.94E+01	8.34E-02	3.58E-02	1.29E-02
1-σ	4.69E+03	8.03E-02	3.04E-02	1.07E-02
e error	2.94E+01	8.32E-02	3.56E-02	1.27E-02
Inclination (i) [deg]	1.96E+00	1.42E-01	7.29E-02	4.67E-02
1-σ [deg]	3.07E+00	1.13E-01	4.83E-02	1.83E-02
error [deg]	1.92E+00	9.97E-02	3.10E-02	4.78E-03

Table A.4

Simulation Results: 5 - Arcsecond Noise Added

Δt [min]	3.83	10	15	25
Ψ_{obs}	6.36E-07	6.96E-06	2.34E-05	1.08E-04
Semimajor Axis (a) [km]	15768	47083	42992	42255
1-σ [km]	4.72E+06	1.09E+06	5.24E+03	1.58E+03
error [km]	2.64E+04	4.92E+03	8.26E+02	8.87E+01
Eccentricity (e)	2.79E+01	1.56E-01	6.04E-02	2.14E-02
1-σ	2.78E+03	2.63E-01	5.41E-02	1.78E-02
e error	2.79E+01	1.56E-01	6.02E-02	2.12E-02
Inclination (i) [deg]	3.17E+00	2.38E-01	1.08E-01	5.50E-02
1-σ [deg]	3.90E+00	2.53E-01	8.18E-02	2.91E-02
error [deg]	3.13E+00	1.96E-01	6.66E-02	1.31E-02

Table A.5

Simulation Results: 10 Arcsecond Noise Added

Δt [min]	3.83	10	15	25
Ψ_{obs}	1.12E-06	7.07E-06	2.34E-05	1.08E-04
Semimajor Axis (a) [km]	37104	29713	45636	42557
1-σ [km]	2.26E+06	5.42E+06	3.40E+05	3.39E+03
error [km]	5.06E+03	1.25E+04	3.47E+03	3.92E+02
Eccentricity (e)	6.37E+01	7.52E+03	1.33E-01	4.33E-02
1-σ	7.05E+03	2.38E+06	1.85E-01	3.72E-02
e error	6.37E+01	7.52E+03	1.33E-01	4.31E-02
Inclination (i) [deg]	4.91E+00	6.80E-01	2.10E-01	8.43E-02
1-σ [deg]	4.60E+00	1.41E+00	1.92E-01	5.81E-02
error [deg]	4.87E+00	6.38E-01	1.69E-01	4.24E-02

Table A.6

Simulation Results: 30 Arcsecond Noise Added

Δt [min]	3.83	10	15	25
Ψ_{obs}	3.25E-06	1.03E-05	2.47E-05	1.08E-04
Semimajor Axis (a) [km]	43604	2680	38758	48167
1-σ [km]	9.70E+06	7.23E+06	3.15E+06	6.52E+05
error [km]	1.44E+03	3.95E+04	3.41E+03	6.00E+03
eccentricity (e)	4.29E+01	4.27E+02	1.04E+01	1.48E-01
1-σ	3.46E+03	1.24E+05	1.10E+03	2.58E-01
e error	4.29E+01	4.27E+02	1.04E+01	1.48E-01
Inclination (i) [deg]	8.10E+00	2.89E+00	1.11E+00	2.32E-01
1-σ [deg]	5.50E+00	3.74E+00	2.11E+00	2.37E-01
error [deg]	8.06E+00	2.85E+00	1.07E+00	1.90E-01