Data Reduction for Diverse Optical Observers Through Fundamental Dynamic and Geometric Analysis

Bradley J. Sease

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Jonathan T. Black, Chair
Craig A. Woolsey
Gregory D. Earle
Brien R. Flewelling

May 21, 2016
Blacksburg, Virginia

Keywords: Space Situational Awareness, Optical Sensing, Orbit Estimation

Data Reduction for Diverse Optical Observers Through Fundamental Dynamic and Geometric Analysis

Bradley J. Sease

(ABSTRACT)

Typical algorithms for processing unresolved space imagery from optical systems make broad assumptions about the expected behavior of the sensors during collection. While these techniques are often successful at data reduction for a particular mission, they rarely extend to sensors in different operating modes. Such specialized techniques therefore reduce the number of sensors able to contribute imagery. By approaching this problem with analysis of the fundamental dynamic equations and geometry at play, we can gain a deeper understanding into the behavior of both stars and space objects viewed through optical sensors. This type of analysis has the potential to enable data collection from a wider variety of sensors, increasing both the quantity and quality of data available for space object catalog maintenance. This dissertation will explore the implications of this approach to unresolved data processing. Sensor-level motion descriptions will be derived and applied to the problem of space object discrimination and tracking. Results of this processing pipeline as applied to both simulated and real optical data will be presented.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The space situational awareness (SSA) problem is an ongoing concern of spacecraft operators. SSA refers to the process of understanding Earth's orbital environment and all of the objects that reside there. Currently, the Space Surveillance Network catalogs over 22,000 Earth-bound objects over 10 cm in size [10], though estimates suggest that there may be hundreds of thousands of objects in the 1 cm to 10 cm size range [11]. Though these objects are small, the high energies involved in the orbital environment cause them to still pose a significant threat to active spacecraft. The current Space Surveillance Network is limited to approximately 30 sensors world-wide [10]. To expand our understanding of Earth's increasingly crowded orbital environment, it is necessary to expand this system and maintain a larger, more precise catalog of space objects.

The focus of this dissertation is the problem of making use of electro-optical image data from diverse sources in support of SSA. The DARPA OrbitOutlook program is one example of a current effort to incorporate more diverse data sources into the SSA pipeline. The aim of the OrbitOutlook program is to reach out to non-traditional sensor operators such as those in commercial, academic, and even hobbyist settings [12]. Though these sensors may not be as precise as the current sensors making up the Space Surveillance Network [10], they are still able to provide valuable data. The inclusion of a larger number of sensors can potentially free up the higher-quality sensors to track higher-priority and hard-to-track objects. Of course, by including low-cost sensors managed by a diverse group of operators, a number of additional challenges arise. Sensors operated at the hobbyist level may often not be tasked in a predictable way, leading to tracking errors and images which are complicated to reduce with current techniques.

To enable analysis of data from of a diverse set of optical sensors, it is necessary to gain a fundamental understanding of the underlying physics and geometry of optical sensor systems. Such an understanding will enable a centralized pipeline that is capable of extracting observations of space objects from data sets including both pristine, well-tracked objects and imperfect, serendipitous detections. Current approaches fall short of providing a centralized pipeline because they typically rely on broad assumptions about the tasking of a sensor. This dissertation focuses on relaxing many of the typical assumptions to create a number of general techniques for reducing optical imagery. Analysis of fundamental projection models in addition to the underlying physics of space objects yields a set of equations describing the motion of a space object across the image plane. These basic equations make minimal assumptions about the sensor motion and enable a number of novel data reduction techniques.

This dissertation is structured as follows.

Chapter 2 will lay out the overall motivation for this dissertation in detail. There are a number of techniques that are currently applied in the field that make broad assumptions about the motion of sensor. While each of these techniques are successful at processing data currently, they each only consider a fraction of the possible operating range of real-world sensors. Further, Chapter 2 will present the necessary mathematical fundamentals required for the following chapters. Specifically, we will consider approaches to modeling the projection equation of an optical observer, image processing for space object extraction, and the principals of orbit determination and estimation.

Chapter 3 derives a series of fundamental equations for objects viewed through an optical sensor. Typical processing procedures rely on only one or two frames for extracting space objects and assume that the sensor follows a specific motion profile. By deriving a set of fundamental equations of motion for objects viewed through a rotating sensor, it is possible to broaden the range of possible sensor motion profiles. Chapter 3 examines this problem in four different ways – two dynamics-based analyses and two geometric analyses. The first approach assumes that a particular optical sensor adheres to the simple pinhole camera model. This model is a common approach and currently used for theoretical analysis of optical navigation in the Earth-Moon system [13, 14]. Since sensors often deviate from the pinhole camera model, we will also consider the case where an image has been corrected in post-processing to follow an "undistorted" projection model. The results of the dynamic approach to image analysis are examined further with a focus on application in Chapter 4.

The geometric analyses focus on leveraging an understanding of the internal and external geometries of an optical sensor to gain additional insight into the resulting image data. The first geometric approach again considers the pinhole camera model and attempts to derive an explicit equation for the path of a star across the image plane. The second geometric method considers the multi-observer case, where multiple observers are viewing a common object. The equations resulting from the geometric approach to image analysis are examined further with a focus on application in Chapter 5.

Chapter 4 looks further into the dynamic approach to image analysis. The dynamic equations of motion from Chapter 2 enable a number of algorithms for detection and tracking of space objects in optical imagery. First, a Kalman filter bank built on the image-plane dynamics allows for robust object tracking and association across frames with a relatively light computational cost. Further, it is possible to use the equations of motion for a star in the image plane to discriminate between stars and non-star objects of interest. Since any space object will have some apparent motion relative to the celestial sphere, a prediction from one frame to the next by equations of motion describing a star will reveal a residual which grows over time. Through careful consideration of the statistics of this residual, we can build a probabilistic argument on whether the object in question is a star or a non-star space object. Further, Chapter 4 shows that it is possible to leverage these motion models for an advanced shift-and-add process, where successive images are shifted to counteract undesired motion in the image plane and subsequently summed together. Prior research has shown that this method can provide a signal gain for dim space objects [15].

Chapter 5 examines the implications of the geometric approach to image analysis. The geometric star path descriptions derived in Chapter 2 offer an alternative way to view image data. A potential problem for space-based optical observers is the fact that precise pointing is not necessarily guaranteed over the course of an observation. In some cases, the spacecraft may drift in such a way as to introduce rotation about the boresight, which further can lead to curvature in observed star signals. These star signals are often critical to measurements of non-star objects of interest as they provide intertially-fixed reference points against which we can determine the sensor's attitude [16, 17, 18]. The geometric path descriptions from Chapter 2 offer a technique for reducing the impact of rotation about the boresight by resampling the image in a polar coordinate frame centered on the point the image about which all the stars rotate.

The multi-observer geometric analyses from Chapter 2 are also revisited in Chapter 5. It is well-known that simultaneous observations from multiple viewpoints offer an immediate solution for the relative position of an object from the sensors [19]. Chapter 5 will prove that observation from multiple, widely-spaced observers not only allows for instantaneous triangulation of an object of interest, but also augments the detection of space objects and allows for probabilistic object associations across sensors.

Finally, Chapter 6 concludes this dissertation with a summary of the work completed and an examination of the practical applications of the work proposed here. Chapters 4 and 5 explore numerous approaches to data reduction for optical observers. These various techniques can be implemented in a variety of combinations, some of which are more applicable to certain end goals. Thus we will conclude by examining some of the combinations of algorithms which are possible and the scenarios in which they are advantageous.

Chapter 2

Background

2.1 Background

The Joint Space Operations Center (JSpOC) processes data from the Space Surveillance Network (SSN), a system of 30 sensors worldwide collecting approximately 400,000 observations per day [10]. The JSpOC currently manages a catalog of over 22,000 Earth-bound objects, including active and inactive spacecraft and debris. Part of the JSPoC's mission, outlined in the United States' National Space Policy, is to make this tracking data available to the public to enable safe operation in an environment which is continually becoming more crowded [20]. Current publicly available catalogs provide data on over 17,000 Earth-bound objects [21]. Estimates of the total debris population in 2001, prior to several notable breakup events, included 330 million pieces between 1 mm and 1 cm in size, 560,000 between 1 cm and 10 cm in size, and 18,000 objects over 10 cm [11].

Catalog data is primarily used for coordinating follow-up observations and conjunction analysis. A continuous understanding of Earth's orbital environment enables prediction of close passes between orbiting objects ("conjunctions") to avoid collisions among the population of active spacecraft and uncontrolled debris. As the orbital environment becomes more crowded, the danger of collisions increases. In the last decade, there are several cases where the debris population has increased dramatically. In 2007, for example, a Chinese anti-satellite test destroyed the decommissioned FengYun 1C spacecraft at a distance of approximately 1,000 km from the Xichang Space Center. This test produced approximately 2,087 pieces of debris which large enough for detection and tracking, and the NASA Orbital Debris Program office estimates there were up to 35,000 pieces of debris as small as 1 cm [1]. The cataloged debris is visualized in Figure 2.1 shortly after impact and 1 year later. Due to variations in perturbations due to size and velocity imparted by the impact, the individual pieces of debris tend to drift apart. After a full year much of the debris is no longer contained in its original orbital plane.

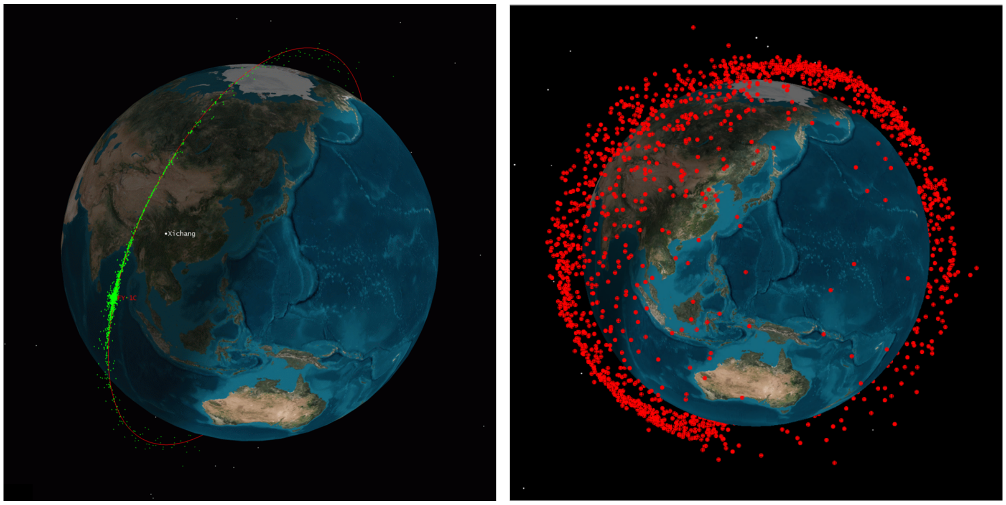

Figure 2.1: FengYun 1C debris after impact (left) and 1 year later (right) [1].

In 2009, the debris population suddenly increased again, this time do to an accidental collision between two spacecraft. Prior to the collision, the active Iridium 33 spacecraft was predicted to pass within 534 meters of the decommissioned Cosmos 2251 spacecraft. The 534 meter pass was not the closest approach predicted that day for an Iridium satellite and the operators took no corrective action. At the time of the expected close approach the Iridium 33 spacecraft ceased communication with the ground and the SSN reported tracking a debris field in its place shortly thereafter. In the aftermath, the SSN cataloged approximately 1,366 pieces of debris belonging to the two spacecraft [2]. The known debris from this collision is visualized in Figure 2.2 at two points in time: 3 hours after impact and 2 years later.

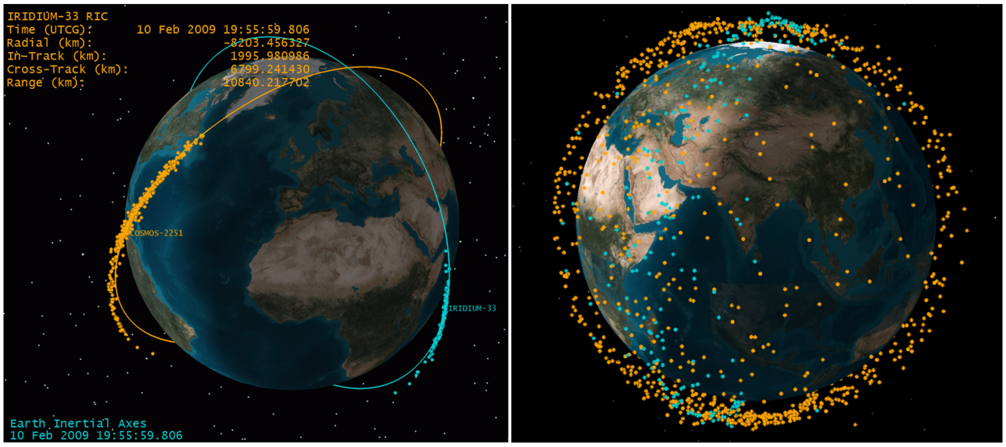
IRIDIUM-33 RIC
Time (UTCG): 10 Feb 2009 19:55:59.806
Radial (km): -8203.456327
In-Track (km): 1995.980986
Cross-Track (km): 6799.241430
Range (km): 10840.217702
COSMOS-2251
IRIDIUM-33
Earth Inertial Axes
10 Feb 2009 19:55:59.806

Figure 2.2: Iridium-Cosmos debris 3 hours (left) and 2 years after impact (right) [2].

With the increasing availability of observation data from diverse sensors, the JSPoC has a need for new hardware and software which can handle the increasing demand and enable the network to make upwards of 40,000 uncorrelated track (UCT) associations per day [22]. In the coming years, the JSPoC expects to handle a volume of greater than 1.5 million observations per day [23]. To handle the sheer volume of data, new techniques are required which can scale up more effectively for future operations.

Though radar systems generate a significant number of the observations that the JSpOC processes each day, the objectives of this proposal are primarily targeted at increasing the quantity and quality of data produced by electro-optical observers. Electro-optical systems typically provide observations of space objects beyond LEO, where the accuracy of radar begins to degrade [11]. Currently, the JSPoC employs the Ground-Based Electro-Optical Deep Space Surveillance (GEODSS) system for tracking objects further than 22,500 km, most notably those inhabiting the Geostationary belt. The JSPoC estimates the this range includes between 4,200 and 4,400 objects [10].

When using electro-optical systems to generate observations of space objects, all signals in an image tend to be unresolved. Stars and space objects have similar appearance, either as points of light or "streaks," where the point-like signal is spread over many pixels. Thus, differentiation between stars and space objects in imagery is non-trivial. There are many different methods of extracting these detections in the literature, which can be separated into two categories: single-frame and multi-frame techniques.

Single-frame techniques extract detections from a single image and typically do not provide any association between frames. Often, sensors are tasked to point at known objects. If the orbital knowledge is precise, then the corresponding space object should appear in a predictable location in the image. In this scenario, the signal closest to the expected location of the space object would be the detection. This type of operating mode is typical for maintenance of existing space object catalogs [24, 25, 26]. Of course, when precise orbital knowledge of an object is not available, a more general detection method is necessary.

Alternatively, single-frame techniques sometimes examine the geometric properties of signals to extract detections. When a space object is imaged against a star field, its motion is often apparent relative to the surrounding stars. With specifically-defined operating procedures it is possible to leverage this fact to discriminate between stars and non-stars from a single image frame. Many existing techniques operate in "sidereal stare" mode, where the sensor slews to track the background star field. In this case, all of the stars are points of light. Any other space object passing through the image, then, will produce a streaked signal that is geometrically unique from the stars. A number of systems have employed these techniques, most notably GEODSS, which provides deep-space observations for the JSpOC [25]. Several space-based observers have leveraged geometric detections as well, including the Space-Based Visible experiment (SBV) [27], the Near Earth Orbit Surveillance Satellite (NEOSSAT) [26], and the Microvariability Oscillations of Stars (MOST) microsatellite[28].

Multi-frame techniques involve analysis of sets of sequential frames to make detections. By considering a longer time frame than a single image, these methods can detect objects which have much lower apparent motion than the single frame techniques. For slow-moving objects, a single frame may not be sufficient to capture its apparent motion. Images taken minutes or hours apart, however, reveal smaller velocities against the stellar background.

The most simple multi-frame methods focus on the data in only two successive images. For example, reference [27] outlines a multi-frame processing scheme in the sidereal star operating mode. When the sensor tracks the stellar background, all stars are static point sources. Subtracting subsequent frames, then, works to remove the star signals. The remaining signals correspond to objects with some apparent motion relative to the background. Frame-to-frame subtraction often leaves some residual of the star signals however, so an alternative approach is to correlate signals between frames. By considering all possible associations exhibiting linear or slightly curved paths, the author in [29] extracts only the objects of interest and ignores star signals. In another application, the Wide Area Space Surveillance System (WASS) considers the variation in individual pixels across frames. Rather than operating in sidereal star mode, the sensor fixes on constant RA/DEC "slices" of the sky to track Geostationary objects. Any pixels which retain signal frame-to-frame correspond to a detection [30].

Several multi-frame applications involve a technique known as "shift-and-add." In shift-and-add operations, several frames are added together in a way that emphasizes certain characteristics of an image. Most commonly, the goal is to move the images with a particular object in such a way as to effectively track the object without moving the sensor. Each sequential image is shifted to counteract the expected motion of a given object. The resulting summation of images should remove the displacement of the target object, while adding additional displacement to all objects which behaved differently. This technique is typically referred to as "synthetic tracking," and has commonly been applied to the detection of near-Earth asteroids [31]. An earlier example of the shift-and-add procedure flew as part of the Space-Based Visible sensor experiment in 1996 to collect spacecraft observations on orbit [32]. Shift-and-add techniques have the additional benefit of increasing the signal-to-noise ratio (SNR) of subsequent detections.

Typically, algorithms like the ones discussed above are developed with specific operational procedures in mind. A single-frame technique based on prior orbital knowledge assumes that the telescope will be operating in either a rate-track mode, where the sensor follows the object, or a step-stare mode, where the sensor points at the location where the object is expected to be. Most other methods expect that the sensor will be fixed in one operating mode throughout collection. The work proposed here aims to derive general techniques that function regardless of the operating mode of the sensor. This research is particularly relevant to the OrbitOutlook program, which aims to take a "data-centric" approach to space situational awareness [12]. Rather than focusing on a limited set of exquisite sensors, the OrbitOutlook program aims to synthesize data from all available sensors from professional assets to hobbyist-level systems. The research outlined in this dissertation provides fundamentally new capabilities for processing data from a diverse set of systems regardless of the operating techniques and with little knowledge of the sensor systems.

A strong motivation for this type of work is the danger posed to active spacecraft by high area-to-mass ratio (HAMR) objects. A European Space Agency survey discovered 28 HAMR objects larger than 10 cm which are suspected to have originated in geosynchronous orbits. This population of objects is estimated to have area-to-mass ratios between 1 m^2/kg and 28 m^2/kg, while a typical spacecraft is around 0.01 - 0.01 m^2/kg. They have concluded that solar pressure has caused these objects to move into highly eccentric orbits [33]. Since this population of HAMR objects crosses through the geosynchronous belt, it follows that they pose a threat to existing spacecraft in that range. Due to the physical properties of these objects – typically pieces of debris – they are highly susceptible to the effects of solar pressure on orbit and tend do deviate widely from the typical orbital models. The difficult to predict nature of the orbits of HAMR objects makes it necessary to take observations frequently [34].

2.2 Mathematical Preliminaries

2.2.1 Modeling the Optical Observer

This section will introduce some of the basic principles of for simulating optical sensor systems. The models described in this section will comprise the foundation for the dynamic analysis in Chapter 2 and the geometric analysis in Chapter 4. Specifically, we will consider two types of models: the pinhole camera model and the distortion-corrected camera model. The pinhole model is a simple representation of an optical sensor and is built on the geometry of the problem and does not consider any other sources of distortion. The second model accounts for distortion in that it is the ideal model that would arise after a sensor's data is perfectly correct for distortion. We will see that these two models are closely related – a fact that will be useful for the derivations in Chapters 2 and 4.

First, we will derive the pinhole camera model. In this model, we assume that an optical sensor allows light in through an idealized aperture with zero extent in any direction. In other words, the aperture is a single point in $\mathbb{R}^3$, referred to as the focal point. The focal plane containing a charge-coupled device (CCD) for digitizing the incoming signal is located at a distance, f, from the focal point. The term f is referred to as the focal length. We assume that light from an object enters the sensor through the focal point and travels in a straight line until it strikes the focal plane. The diagram in Figure 2.3 shows a 2-dimensional representation of this geometry. In this figure, Δx refers to the displacement of the point of intersection from the center of the CCD. The diagram for the displacement along the second dimension of the CCD would be identical, replacing Δx with Δy.

With the pinhole camera in mind, we wish to derive a mathematical representation of the geometry. This set of equations will be the foundation for the analysis in Chapter 2. Consider the diagram in Figure 2.4. This is the same setup as in Figure 2.3, with the addition of two

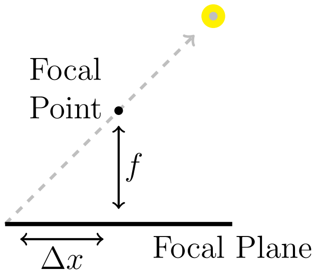

Figure 2.3: Two dimensional diagram of the pinhole camera model.

more terms: ρ_1 and ρ_3. Define $\boldsymbol{\rho} = [\rho_1, \rho_2, \rho_3]^T$ to be the location of a particle in $\mathbb{R}^3$ that is in view of the sensor. In this diagram, the lower triangle defines the geometry of the projected particle internal to the sensor, while the upper triangle defines the positioning of the particle relative to the focal point of the sensor.

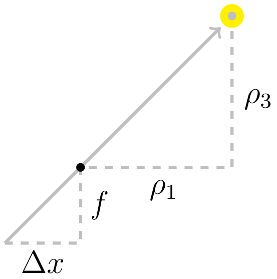

Figure 2.4: Vector geometry of the pinhole camera model.

Since these two triangles are similar (i.e. all angles are the same) the proportions between corresponding sides are the same. Thus, we can say that

$$\frac{\Delta x}{f} = \frac{\rho_1}{\rho_3} \tag{2.1}$$

The equation above relates a particle's location in $\mathbb{R}^3$ to its displacement in one direction from the center of the CCD. Multiplying both sides by the focal length yields

$$\Delta x = \frac{f}{\rho_3}\rho_1 \tag{2.2}$$

An identical derivation follows for Δy and ρ_2 to get

$$\Delta y = \frac{f}{\rho_3}\rho_2 \tag{2.3}$$

To proceed further, we must define a coordinate system in which we will represent the position of a particle in the image plane. Consider the coordinate system denoted by $\boldsymbol{o}$ in

Figure 2.5. Using this coordinate system, we can write

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\frac{f}{\rho_3} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{2.4}$$

The relation above is called the Collinearity equation [35]. The underlying assumption behind this model is that, as seen in Figure 2.3, a particle in $\mathbb{R}^3$ and its $\mathbb{R}^2$ image plane projection are connected by a line passing through the focal point of the sensor. Note that this is a purely physical model at this point, so x and y are quantified with units of distance. This means that there is an additional step required to convert from the physical units to the pixel grid that is typical of digital image data. For the purposes of this dissertation, the origin for the pixel grid is denoted by $\bar{\boldsymbol{o}}$ in Figure 2.5. The relationship between $\boldsymbol{o}$ and $\bar{\boldsymbol{o}}$ is

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \frac{1}{s} \begin{bmatrix} x \\ y \end{bmatrix} + \frac{1}{2} \begin{bmatrix} r_x \\ r_y \end{bmatrix} \tag{2.5}$$

where $\boldsymbol{r} = [r_x, r_y]^T$ is the number of physical pixels on the CCD in the x and y directions and s is the physical width of a single pixel. In the new scaled coordinates, $(\bar{x}, \bar{y})$ gives the location of of a particle projected onto a CCD, measured in pixels. So, a particle located directly along the boresight of the sensor (i.e. the sensor is pointing at the particle) is located at $(\bar{x}, \bar{y}) = ({}^{r_x}/_2, {}^{r_y}/_2)$. Note that this equation and the diagram in Figure 2.5 assumes that pixels are square.

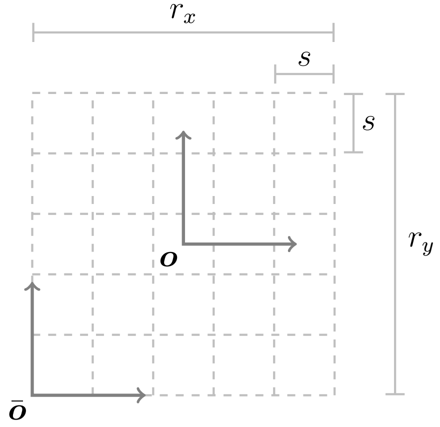

Figure 2.5: Focal plane coordinate systems and dimensions.

Equally important to the forward projection model for the pinhole camera is the inverse projection from $\mathbb{R}^2$ to $\mathbb{R}^3$. Consider again Figure 2.3. Note that the line of site vector (the gray, dashed line) is the same for any object along that line, regardless of distance from the observer. In fact, the range to a particular object is unobservable from a single image-plane point in a single sensor. In resolved imagery, where the observer is close enough to an object to make out details on its surface, it is sometimes possible to calculate range from a single observer. For example, a currently-studied approach for optical navigation uses imagery of the moon to ascertain a spacecraft's absolute position in the Earth-Moon system [14]. This dissertation will focus on the case of "unresolved" imagery, where it is not possible to discern any of the detail of an object in image and the object's range is unobservable. It is possible, however, to determine the line of sight vector of the particle in Figure 2.3

The internal geometry of the line of sight vector (from the focal plane to the focal point) is given by $[x, y, f]^T$. Since the magnitude of this vector does not correspond to the distance of the object from the sensor, we are only interested in the direction. Therefore we write the line of sight vector as

$$\boldsymbol{u} = \frac{\boldsymbol{\rho}}{\|\boldsymbol{\rho}\|} = \frac{1}{\sqrt{x^2 + y^2 + f^2}} \begin{bmatrix} x \\ y \\ f \end{bmatrix} \tag{2.6}$$

Finding the physical focal plane coordinates (x, y) from the pixel coordinates $(\bar{x}, \bar{y})$ requires a coordinate system shift from $\bar{\boldsymbol{o}}$ to $\boldsymbol{o}$. This shift is given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = s \begin{bmatrix} \bar{x} - {}^{r_x}\!/_2 \\ \bar{y} - {}^{r_y}\!/_2 \end{bmatrix} \tag{2.7}$$

Thus an image-plane measurement of a signal provides a unit vector describing the relative position of the corresponding object relative to the observer. The object's range is unavailable, but it is located at some distance along the line of sight vector $\boldsymbol{u}$. It is sometimes beneficial to express the line of sight vector as a pair of angles. In the sensor body frame, we can use the azimuth and elevation, which are related to $\boldsymbol{u}$ through [36]

$$\begin{aligned} \mathrm{AZ} &= \tan^{-1}\left(\frac{u_2}{u_1}\right) \\ \mathrm{EL} &= \tan^{-1}\left(\frac{u_3}{\sqrt{u_1^2 + u_2^2}}\right) \end{aligned} \tag{2.8}$$

If the line of sight vector is expressed in the Earth-Centered Inertial coordinate system, then the azimuth and elevation correspond directly to right-ascension and declination [36].

The pinhole camera model described above accounts for the idealized geometry of an optical sensor. This model does not, however, account for any other sources of distortion. Often, distortion represents a significant deviation from the ideal sensor. A number of standards exist for approaching the problem of correcting an image to account for distortion [37, 38]. The Simple Image Polynomial (SIP) distortion representation, for example, uses two distortion polynomials to convert between pixel coordinates and "world coordinates" [37]. This representation is an alternative to the inverse projection from Equation 2.6 that accounts for imperfections in the sensor geometry. The SIP correction from pixel coordinates to intermediate world coordinates is given by [37]

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} \bar{x} + f\left(\bar{x}, \bar{y}\right) \\ \bar{y} + g\left(\bar{x}, \bar{y}\right) \end{bmatrix} \tag{2.9}$$

where $f(\bar{x}, \bar{y})$ and $g(\bar{x}, \bar{y})$ are distortion polynomials. These polynomials are standardized in the form

$$\begin{aligned} f(x, y) &= \sum_{p,q} A_{p,q} x^p y^q, \qquad p + q \leq n \\ g(x, y) &= \sum_{p,q} B_{p,q} x^p y^q, \qquad p + q \leq m \end{aligned} \tag{2.10}$$

where n is the order of f and m is the order of g. Note that it is up to the user to choose the precise model for the correction process. The standards do not specify any target model that the corrected coordinate system must approximate. Instead, the standards remain general enough to accommodate any set of world coordinates [37]. The conversion from the intermediate world coordinates $(\hat{x}, \hat{y})$, to the desired world coordinates is [37]

$$\begin{bmatrix} \alpha \\ \delta \end{bmatrix} = \underbrace{\begin{bmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{bmatrix}}_{C_d} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \tag{2.11}$$

where (α, δ) are the desired world coordinates. Derivation of the distortion polynomials is possible from a background star field. The Astrometry.net star identification software package returns a set of second order SIP polynomials that convert from pixel space to right ascension and declination [18].

In addition to the projection equations it is sometimes also necessary to model the expected intensity of a signal in the final image. The brightness of a space object is typically measured on a log scale. In [7], the author provides a relationship describing the intensity of a signal when viewed for a particular sensor. The electronics of the sensor effectively count the number of incoming photons with some efficiency. Photons that have been successfully counted by the sensor are referred to as photoelectrons [7]. The estimated signal intensity for a particular source is

$$I = \frac{m_0 a \Delta t}{2.5^{(m_v)}} \tag{2.12}$$

where m_0 is the number of photoelectrons per second per square meter of aperture received by the sensor for a zero magnitude object, a is the aperture area, Δt is the exposure time, and m_v is the visible magnitude of the object in question. Based on the equation above, we see that as the visible magnitude, m_v, becomes smaller, the signal intensity increases. This is due to the fact that the visible magnitude is represented in the exponent of the denominator of the equation above. A negative visible magnitude moves the denominator into the numerator and acts as a multiplier. The Sun, for example, has a visible magnitude of -26.7, far brighter than any other object in the sky [7]. Experimentally-derived results in [7] give a value of $m_0 = 19,100$ photoelectrons per second per square millimeter of aperture area for a magnitude zero star, however this value will vary with each sensor. Further, the signal captured by a sensor will also vary due to various noise inputs.

Optical sensors contain three different categories of noise: shot noise, read noise, and dark current [39]. The first, shot noise, is due to variation in the number of incoming photons from the source. This variation may be due to atmospheric effects or changes in the object itself. Shot noise typically follows a Poisson distribution and increases in variance with the signal strength [39]. Read noise is introduced by the internal electronics of the sensor. This noise input is independent of the exposure time and signal strength and is simply due to noise in the circuitry during read-out. Typical levels of read noise are on the order of about 10 photoelectrons [39]. The third noise input is due to current leakage in the internal circuitry. Dark current is proportional to the exposure time and may vary in intensity over the chip. Typical levels of dark current noise at room temperature are about 2.5×10^4 photoelectrons per pixel per second but can be as low as 0.04 photoelectrons per second for cooled sensors [39]. Figure 2.6 illustrates these different types of noise. The read noise and dark current are dominant in dark areas of the image while the shot noise dominates brighter areas.

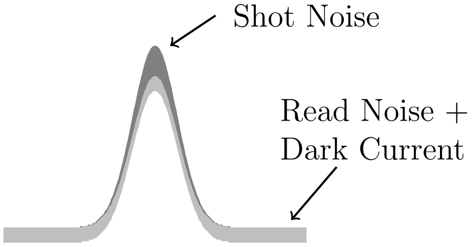

Figure 2.6: Illustration of the effect of types of image noise on a point-source signal.

2.2.2 Image Processing for Unresolved Space Imagery

Now we will examine the problem of extracting objects from unresolved images of space objects. There are a number of approaches for extracting space objects from optical imagery. The most popular method is a software package called Source Extractor, which aims to extract space objects from general imagery with no knowledge of sensor characteristics [40]. Most approaches described in the literature rely on simple tools such as

Before beginning, let us define some of the conventions we will follow for the remainder of this dissertation. Images are captured and stored as discrete pixel data, however to simplify notation we will consider images as a continuum. Unless otherwise noted, the term $I(x, y)$ will represent an image intensity map. This corresponds to a grayscale image where $I(x, y)$ is the intensity of a theoretical pixel centered on the coordinate (x, y). To convert to the discrete representation, we simply only evaluate I at the discrete (x, y) locations corresponding to physical pixels on the CCD.

The most basic method for separating signal from background noise is a thresholding process. With a thresholding process, the goal is to build a binary mask which indicates which pixels contain signal and which contain only noise. We achieve this by choosing a signal level at which a pixel is considered to contain a signal and then examining each pixel individually. The image mask is then

$$M\left(x,y\right)=\begin{cases}1 & \text{if} \quad I\left(x,y\right)>T \\ 0 & \text{if} \quad I\left(x,y\right)\leq T\end{cases} \tag{2.13}$$

where $M\left(x,y\right)$ denotes a binary image mask and T is some threshold level. There are many ways to choose the correct threshold level, but the most common is through analysis of a histogram of the image intensity data [41]. Figure 2.7 contains a simulated image to demonstrate the thresholding concept. The noisy image on the left-hand side of the figure contains a number of stars and a single space object passing through the image. Homogeneous, normally-distributed noise corrupts each pixel individually. The right-hand side is the ideal mask where all of the pixels containing signal are white and all other pixels are black.

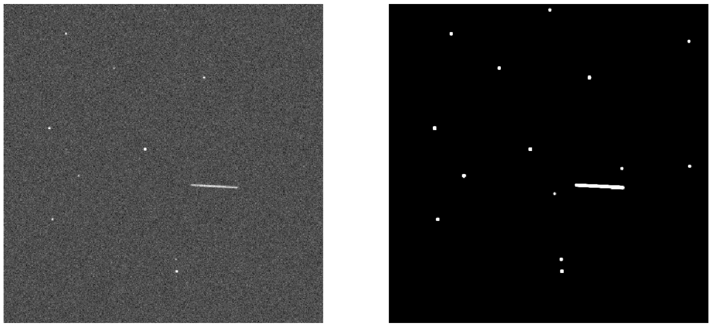

Figure 2.7: Simulated noisy star image (left) and its ideal image mask (right).

Figure 2.8 shows the noisy image from Figure 2.7 thresholded at various levels. The first image is limited to only signals at 2σ above the mean. This result removes a significant portion of the noise, but there are a large number of single pixels that were above the threshold and now clutter the mask. The center image in the figure corresponds to a threshold of 3σ above the mean. This result is much cleaner, but still contains some clutter from noise. Finally, the third image in Figure 2.8 is the result of a threshold at 4σ above the mean. This result does not contain any clutter from noise, but comparing it to the ideal mask in Figure 2.7 shows that a number of stars fell entirely below the threshold level. So in this case a threshold process requires an object to have a signal to noise ratio (SNR) of 3 - 4 for a successful detection. Dimmer objects will be lost.

A number of image processing techniques rely on gradient-based approaches to understanding the image data. These methods compute the spatial image intensity gradients across a pixel in order to characterize the behavior inside that pixel [42]. This approach allows one to infer the contents of a particular pixel (i.e. noise or a signal) to move onto higher-level processing. The gradient-based techniques that we will consider here search for large gradients in one or two directions to indicate the presence of a key feature at that location. These methods can also be used to detect features in unresolved imagery.

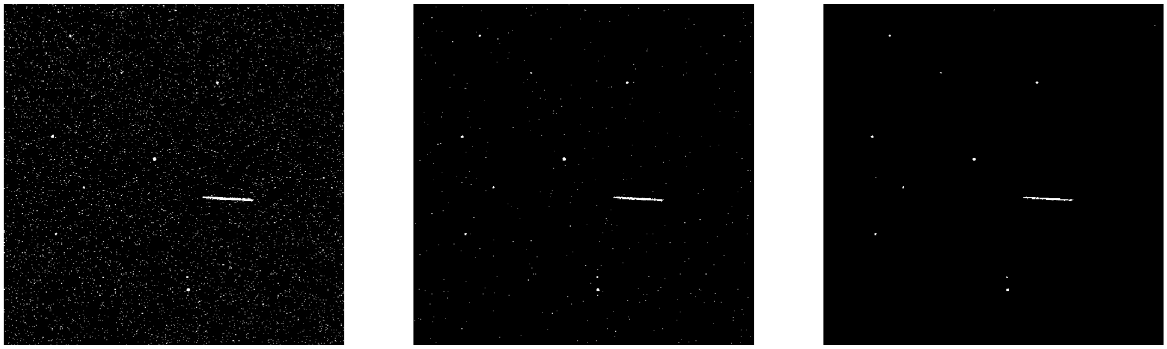

Figure 2.8: Simulated thresholding at 2 sigma (left), 3 sigma (center), and 4 sigma (right).

Before we formulate the object detection routine, we must first define the image gradients. These gradients are derivatives of the signal intensity taken spatially across the image. Since image data is discrete, we must approximate the derivatives in an approach similar to a finite difference. Most commonly the spatial derivatives are taken with a derivative kernel that is convolved with the original image data [42]. For example, 3 × 3 derivative kernels in the x and y directions are

$$\boldsymbol{K}_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{K}_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.14}$$

Subtraction of neighboring pixels computes the change in intensity over that pixel which directly corresponds to the finite difference of the intensity at that pixel. Formally, the derivatives of the image intensity data, $I\left(x,y\right)$, are

$$\frac{\delta I\left(x,y\right)}{\delta x} \approx \boldsymbol{K}_x * I\left(x,y\right), \qquad \frac{\delta I\left(x,y\right)}{\delta y} \approx \boldsymbol{K}_y * I\left(x,y\right) \tag{2.15}$$

Like a standard finite difference, these derivative kernels tend to be highly susceptible to the presence of noise in the individual pixels. It is beneficial, then, to smooth the image to reduce the effect of noise. We can achieve this smoothing by convolving the image with a Gaussian kernel. The Gaussian kernel is [42]

$$G\left(x,y,\sigma\right) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2.16}$$

So, the smoothing process is written as

$$I_s\left(x,y\right) = G\left(x,y,\sigma\right) * I\left(x,y\right) \tag{2.17}$$

where $I_s\left(x,y\right)$ is the smoothed intensity map. The most common gradient-based methods use these derivatives to determine the principal gradients at each pixel. It is possible to infer the presence of a particular feature at that pixel based on the principal gradients. To compute the principal gradients, we use a matrix called the structure tensor, $\boldsymbol{Z}(x, y)$ [43].

$$\boldsymbol{Z}(x, y) = \begin{bmatrix} \left(\frac{\delta^2 I}{\delta x^2}\right)^2 & \frac{\delta I}{\delta x}\frac{\delta I}{\delta y} \\ \frac{\delta I}{\delta x}\frac{\delta I}{\delta y} & \left(\frac{\delta^2 I}{\delta y^2}\right)^2 \end{bmatrix} \tag{2.18}$$

Note that in the equation above the image intensity is shown without the function inputs for compactness, but it remains a function of x and y. This matrix, evaluated at a particular pixel, provides information on the nature of the data in that pixel. The first method for determining the nature of each pixel relies on analysis of the eigenvalues of $\boldsymbol{Z}$. Two large eigenvalues implies that two strong gradients cross over that pixel and therefore it is considered to contain a corner. One large eigenvalue indicates a large gradient in a single direction which further suggests that the pixel contains an edge. If both eigenvalues are small, then the region is considered flat [43]. So, to create a response that emphasizes edges in the original image, one should take the maximum eigenvalue of the structure tensor for each pixel. Formally,

$$r_{\text{edge}}(x, y) = \max\left\{\lambda_1(x, y), \lambda_2(x, y)\right\} \tag{2.19}$$

If instead we wish to create a response that highlights corners in the original image data, the minimum eigenvalue at each pixel provides this result.

$$r_{\text{corner}}(x, y) = \min\left\{\lambda_1(x, y), \lambda_2(x, y)\right\} \tag{2.20}$$

An alternative method aims to achieve the same performance as the eigenvalue-based approach with a lower computational cost. The combined corner and edge detector proposed by [44] uses the structure tensor but does not compute the eigenvalues. Instead, the algorithm computes a combined corner and edge response, r_h, which is [44]

$$r_h(x, y) = \left(\frac{\delta^2 I}{\delta x^2}\frac{\delta^2 I}{\delta y^2}\right)^2 - \left(\frac{\delta I}{\delta x}\frac{\delta I}{\delta y}\right)^2 - k\left(\frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2}\right)^2 \tag{2.21}$$

or, in terms of $\boldsymbol{Z}$,

$$r_h(x, y) = \det(\boldsymbol{Z}) - k\ \mathrm{tr}(\boldsymbol{Z})^2 \tag{2.22}$$

In this equation, k is a user-defined tuning parameter [44]. When r_h is positive, the pixel contains a corner. If r_h is negative, then the pixel contains an edge. Whenever the response, r_h, is small, the region is considered to be flat.

Another edge and corner detector analyzes an image in frequency space rather than the raw intensity data. Phase congruency, a measure of the amount of agreement in the fourier components of a signal, provides an indicator for the type of features that a pixel contains. The phase congruency for a one-dimensional signal is [45]

$$PC(x) = \frac{|E(x)|}{\sum_{i=1}^{n} A_i(x)} \tag{2.23}$$

where $E(x)$ is the sum of the components of the Fourier transform and $A_i(x)$ is the amplitude of the ith Fourier component at a particular location, x. The numerator of the equation above is the resultant of the sum of all of the Fourier components at that location, referred to as the "local energy." Regions where the Fourier components are mostly in phase will have a high phase congruency while regions where the Fourier components are not aligned will return a low result. Figure 2.9 illustrates this sum of Fourier components. The maximum value of this ratio is 1, when all of the component vectors perfectly aligned.

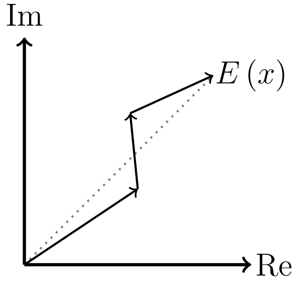

Figure 2.9: Example phase congruency local energy plot.

Equation 2.23 only considers a case in one dimension. The work in [45] computes a value for phase congruency along a number of lines crossing a pixel at a specified angle in order to examine the principal moments of the phase congruency at that point. We write the phase congruency along a line crossing a pixel at an angle θ to the horizontal as $PC(x, y, \theta)$. If we also define:

$$\begin{aligned}\nu_1(x,y) &= \sum_\theta [PC(x,y,\theta)\cos\theta]^2 \\ \nu_2(x,y) &= 2\sum_\theta [PC(x,y,\theta)\cos\theta][PC(x,y,\theta)\sin\theta] \\ \nu_3(x,y) &= \sum_\theta [PC(x,y,\theta)\sin\theta]^2\end{aligned} \tag{2.24}$$

Then the phase congruency edge response is

$$r_{pc} = \frac{1}{2}\left(\nu_3(x,y) + \nu_1(x,y) + \sqrt{\nu_2(x,y)^2 + [\nu_1(x,y) - \nu_3(x,y)]^2}\right) \tag{2.25}$$

and the phase congruency corner response is

$$r_{pc} = \frac{1}{2}\left(\nu_3(x,y) + \nu_1(x,y) - \sqrt{\nu_2(x,y)^2 + [\nu_1(x,y) - \nu_3(x,y)]^2}\right) \tag{2.26}$$

The edge and corner responses correspond to the maximum and minimum phase congruency values for any angle θ. These results are analogous to the minimum and maximum gradients in the eigenvalue-based approach. Figure 2.10 shows a real-world example of the benefits of edge detection methods for processing unresolved imagery. The image on the left-hand side of the figure is an image of the comet Pan-Starrs taken by a hobbyist astronomer [3]. The comet's coma in the original image introduces a gradient across a portion of the image that would complicate analysis with a simple threshold process. The image on the right-hand side is the result of the phase congruency edge detector run on the original image. We see that in the result the star streaks are emphasized and the effect of the noise has been reduced. Further, the edge detection routine completely ignored the comet's coma, leaving only a small point source indicating the location of the comet in the image.

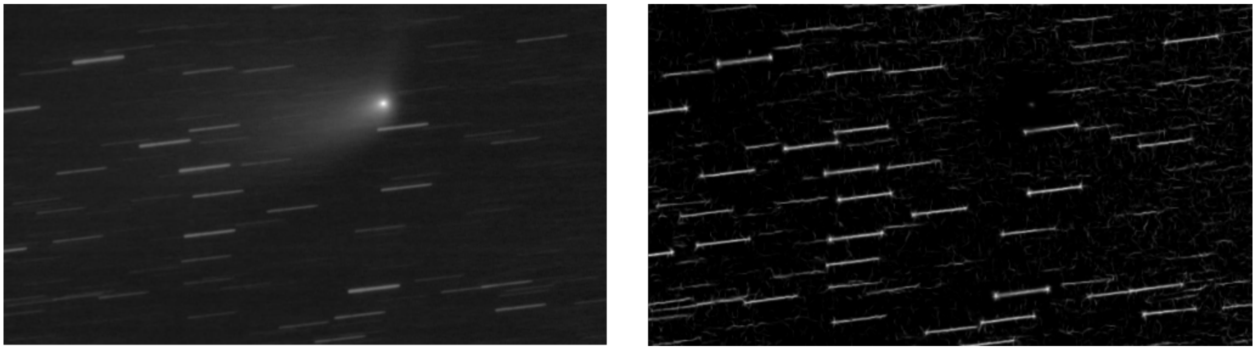

Figure 2.10: Result of an edge detection process on an image of the comet Pan-Starrs [3].

Past work has also shown that the corner detector is an approximate method for extracting the endpoints of streaked point sources. Since streaks typically have a consistent gradient along the direction of travel, the only points at which they exhibit a strong bi-directional gradient is at the ends. See [46] for a complete discussion of this approach to star streak analysis.

A drawback of threshold and gradient based approaches is their interaction with image noise. In order to detect objects near to or below the noise floor, it is often necessary to employ some prior knowledge to extract the signals. Matched filter analysis is a technique that is commonly used for detection of dim near-Earth asteroids [47] and spacecraft [24]. The basic concept behind matched filter analysis is that, if we know what an object is expected to look like in optical imagery, then we can leverage that knowledge to aid in extracting its location even when it is too dim for traditional methods. We achieve a signal gain by building a template that describes the object's expected appearance in the image based on some prior knowledge or initial guess. In the next chapter we will derive motion models for objects across the image plane of an optical sensor and these models will provide a way of predicting the geometry of signals of interest.

However we create this template, whether by assuming a velocity for the object [47] or through prior knowledge of a particular space object [24], we can leverage it to extract objects that match the template through convolution. Formally, the convolution process is

$$R\left(x,y\right)=T\left(x,y\right)*I\left(x,y\right) \tag{2.27}$$

where $T(x, y)$ is the template image and $R(x, y)$ is the matched filter result. A peak in the resulting image suggests that there may be a matching signal at that location. Figure 2.11 contains a simulated example of this matched filter process. The image on the left-hand side of the figure is a simulated noisy star image. The right-hand side of the image contains the result of convolution of the original image with a matched filter template. In this case we assume that all of the stars have an identical appearance. In the result we see that even the dim star streaks show a significant gain relative to the background noise.

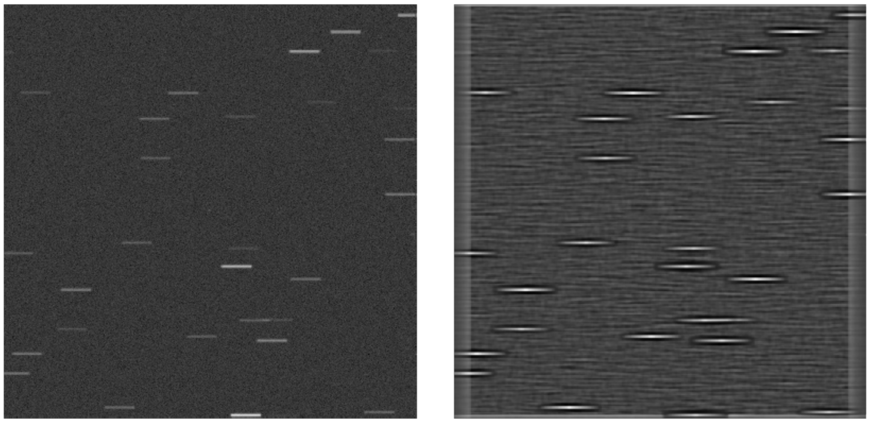

Figure 2.11: Simulated star image (left) and the result of a matched filter convolution (right).

The matched filter analysis described above has the benefit of simultaneously providing a location for the detection as well. Methods based on thresholding or edge detection require a secondary process for localization of objects in an image. The most common method for localizing objects in unresolved imagery is centroiding. Similar to the centroid of a physical object, the centroid in an image computes a "center of intensity" that consists of a weighted sum of all of the pixels that make up a particular object. This process is capable of providing precise estimates of the location of an unresolved signal in an image and the result actually improves for unfocused optics [48, 7]. The centroid calculation can be written as

$$\boldsymbol{c} = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i \; I(\boldsymbol{x}_i)}{\sum_{i=1}^{n} I(\boldsymbol{x}_i)} \tag{2.28}$$

where n is the number of pixels contained in the signal of interest, $\boldsymbol{x}_i$ is the location of the ith pixel, and $I(\boldsymbol{x}_i)$ is the intensity value of that pixel. Accuracies as low as a hundredth of a pixel are possible for bright, static objects [7]. The centroid process suffers when objects begin to streak across the image. As the speed of an object across the image plane increases, its signal is spread across more pixels. This spread reduces the signal brightness and increases the effect of noise, thereby reducing the accuracy of the centroid process [49].

A slightly different approach for computing an object's centroid can sometimes provide improved accuracy. The mean shift process iteratively searches for a local maximum through a sequential centroiding routine [50]. Rather than including all of the pixels that are considered to be a part of the object, some of which might be noise, the mean shift process chooses a small window of the image and iteratively computes the centroid of that window. For each iteration, the algorithm computes the center point, then shifts the window to that center point for the next iteration. Formally, the centroid of a window with radius r is

$$\boldsymbol{c}_w = \frac{1}{w} \sum_{i=-r}^{r} \sum_{j=-r}^{r} \left\{ \left(\boldsymbol{x}_p + \begin{bmatrix} i \\ j \end{bmatrix} \right) I\left(x_p + i, y_p + j\right) \right\} \tag{2.29}$$

where $\boldsymbol{x}_p = (x_p, y_p)$ is the center point of the window and

$$w = \sum_{i=-r}^{r} \sum_{j=-r}^{r} I\left(x_p + i, y_p + j\right) \tag{2.30}$$

This equation is equivalent to Equation 2.28 when considering only the pixels within a prescribed window. At each iteration, one can either round $\boldsymbol{x}_p$ so that it falls only in the center of a pixel or interpolate values at subpixel locations in $I\left(x_p + i, y_p + j\right)$. The mean shift, or the shift from one iteration to the next, gives an indication of the convergence of the algorithm. The shift is defined by [50]

$$\boldsymbol{m}_i = \boldsymbol{c}_w^{[i]} - \boldsymbol{c}_w^{[i-1]} \tag{2.31}$$

where $\boldsymbol{c}_w^{[i]}$ indicates the centroid computed from the ith iteration. So, one choice of convergence criteria is the euclidean norm of the mean shift. When this error metric drops below a defined threshold, τ, then the algorithm will stop iterating.

$$\epsilon_i = \left\|\boldsymbol{m}_i\right\|_2 \leq \tau \tag{2.32}$$

Figure 2.12 contains a simulated example of this process in a noise-free scenario. The algorithm begins with only a pixel-accurate estimate of the centroid of a point source which is approximately 0.34 pixels from the true, subpixel center point. By the third iteration, the error has dropped below a tenth of a pixel. Beyond the 6th iteration, the error is below a hundredth of a pixel.

Many of the algorithms described thus far suffer from the presence of image noise. There are, however, a number of approaches for reducing and removing the effects from predictable noise patterns [39, 51, 52]. These techniques work by characterizing the sensor either offline or from a single image to predict the effect of various noise sources in a single pixel. This prediction then allows for subtraction of some or all of that noise source.

As mentioned previously, dark current is a source of noise that is proportional to the exposure time of an image. Since it is proportional to the exposure time, we may attempt to estimate the effect of this noise source and counteract it. A common method called dark frame subtraction works to create an image with only dark current noise to be subtracted from

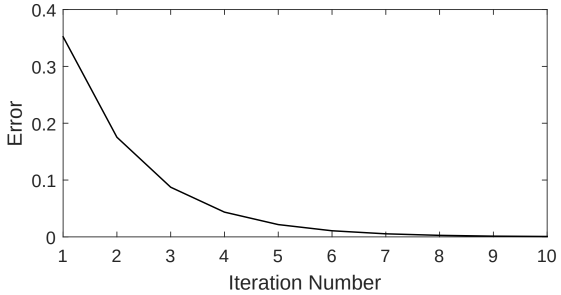

Figure 2.12: Mean shift convergence from pixel-accurate initial estimate.

later image data. To create a dark frame, we capture an image with the lens cap of the sensor on. This way there is no source of light, so the only signal should be due to noise. If we expose for a sufficient duration the dark current noise will dominate the read noise and it should be possible to build an accurate dark frame. Once the dark frame is built, it is then possible to subtract it from subsequent imagery in attempt to remove some of the effect of this noise source [39, 51]. In general, dark frame subtraction is able to reduce the effect of dark current noise, however it does not reliably benefit detection of dim space objects [51].

Another source of error in imagery is due to variation in the efficiency with which physical pixels on the CCD convert from photons to photoelectrons. Ideally, two pixels illuminated with the same photon flux would produce identical values. In practice, however, the CCD will contain small variations of efficiency between individual pixels [39]. To correct these spatial sensitivity variations, we can extract a flat field map and reverse the effect. To create a flat field map one would ideally uniformly illuminate the CCD so that every pixel is receiving an identical photon flux. The resulting image would, in a perfect sensor, contain uniform values over the entire space. Since the sensor is not perfectly homogeneous, however, there will be regions which are brighter or dimmer than average. Manipulating the original image such that each pixel has a uniform signal in the flat field map will work to counteract this effect [39].

Some image corruption is due to actual, undesired signals. For example, stray light entering the aperture or overly-bright objects can cause glare that obscures large portions of the image. An algorithm for reducing this effect involves analysis of the actual image data. If we are able to determine which pixels are background and which pixels contain signal, then it is possible to perform a background fit to subtract whatever gradients exist over the relevant image data [52]. Specifically, we can fit a polynomial to each column (or row) to describe the contribution of "background noise" to each pixel. For each column, define the background polynomial $b\left(y\right)$,

$$b\left(y\right) = \alpha_n y^n + \alpha_{n-1} y^{n-1} \ldots + \alpha_1 y + \alpha_0 \tag{2.33}$$

where n is the order of the polynomial and α_i for $i = 0, \ldots, n$ are unknown constants. If we are able to find ℓ pixels which are part of the background and do not contain any signal, then we can build up the system of equations

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{\ell-1} \\ b_\ell \end{bmatrix} = \begin{bmatrix} y_1^n & y_1^{n-1} & \dots & y_1 & 1 \\ y_2^n & y_2^{n-1} & \dots & y_2 & 1 \\ & & \vdots & & \\ y_{\ell-1}^n & y_{\ell-1}^{n-1} & \dots & y_{\ell-1} & 1 \\ y_\ell^n & y_\ell^{n-1} & \dots & y_\ell & 1 \end{bmatrix} \begin{bmatrix} \alpha_n \\ \alpha_{n-1} \\ \vdots \\ \alpha_1 \\ \alpha_0 \end{bmatrix} \tag{2.34}$$

A least squares solution for $(\alpha_0, \dots, \alpha_n)$ will characterize the background noise in that particular column. By building a complete background image from this process, we can then attempt to subtract its effect and flatten the image. Figure 2.13 contains a real image on which we will demonstrate this effect. The image contains the star Alpha Centauri which is bright enough to cast a glare over a large portion of the image, obscuring a number of dimmer stars [53]. We wish to perform a background subtraction to flatten the image out and remove as much of the glare as possible. For this particular image, separating the background from the signals is difficult, so instead we will simply fit a polynomial to each column. After the initial fitting process, we will remove any pixel which is brighter than the background and perform a secondary fit.

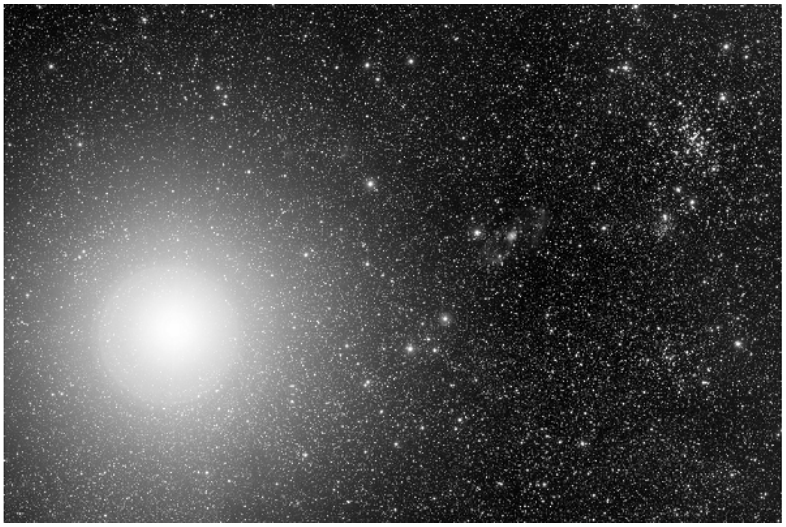

Figure 2.13: An image of Alpha Centauri with a glare obscuring neighboring stars.

Figure 2.14 contains the result of this two-step background subtraction process. We see that the bulk of the glare from Alpha Centauri has been removed, though some artifacts remain in the region around the star. For many of the neighboring stars, though, this glare reduction offers a considerable improvement.

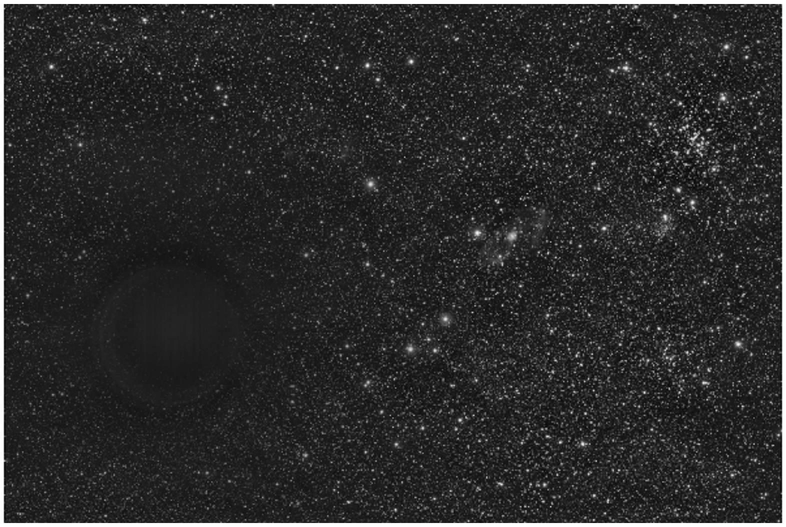

Figure 2.14: Result of a background subtraction process on the Alpha Centauri image.

The image processing approaches detailed in this section make up the foundation of the analysis in Chapters 4 and 5. Specifically, these techniques work to extract space objects from unresolved imagery. The derivations in Chapters 4 and 5, for the most part, operate on the assumption that precise locations are available for all of the space objects in an image. So, with the methods described here we are able to extract measurements in the form of (x, y) locations to drive higher-level analysis in the coming chapters.

2.2.3 Orbit Determination and Estimation

The bulk of this dissertation will focus on approaches for taking optical measurements of space objects. The ultimate goal of these observations, however, is to determine the orbit of an observed object. Achieving this goal requires that we determine a position vector, $\boldsymbol{r}$, and a velocity, $\dot{\boldsymbol{r}}$, that describe the object at a given time. There are a number of ways to do this, only a subset of which will be discussed here. Before discussing the orbit determination and estimation processes it is necessary to consider the equations that govern the motion of orbiting bodies. There are many approaches to model orbital motion, each considering different sets of forces. In general, we can express orbital motion by the following second order differential equation [36].

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a}_{\text{NonSpherical}} + \boldsymbol{a}_{\text{AirDrag}} + \sum_{i=1}^{n} \boldsymbol{a}_{\text{ThirdBody}}^{[i]} + \boldsymbol{a}_{\text{SolarPressure}} \tag{2.35}$$

Here, $\boldsymbol{r}$ describes the position of a spacecraft relative to the center of the body that it is orbiting. The term μ is the gravitational parameter. For Earth, $\mu \approx 398600$ km^3/s^2. The first term of this equation describes the orbit of a spacecraft when the central body is expressed as a point mass and all other forces are neglected. The second term, $\boldsymbol{a}_{\text{NonSpherical}}$, accounts for variations in the gravity field of Earth due to deviations from a homogeneous, spherical shape. The third term, $\boldsymbol{a}_{\text{AirDrag}}$, describes the effect of drag from the upper atmosphere on the spacecraft.

$$\boldsymbol{a}_{\text{AirDrag}} = -\frac{1}{2} C_D \rho \frac{A}{m} \frac{\dot{\boldsymbol{r}}_{\text{rel}}}{\|\dot{\boldsymbol{r}}_{\text{rel}}\|} \dot{\boldsymbol{r}}_{\text{rel}}^2 \tag{2.36}$$

where C_D is the coefficient of drag, ρ is the local atmospheric density, A is the cross-sectional area of the spacecraft, m is the mass of the spacecraft, and $\dot{\boldsymbol{r}}_{\text{rel}}$ is the velocity of the spacecraft relative to the atmosphere (which is rotating). The relative velocity is [36]

$$\dot{\boldsymbol{r}}_{\text{rel}} = \frac{d\boldsymbol{r}}{dt} \left(-\boldsymbol{\omega}_{\oplus} \times \boldsymbol{r}\right) \tag{2.37}$$

It is important to note that precisely determining the local atmospheric pressure, ρ, is non-trivial. The local density is affected by a number of time-varying parameters such as weather patterns, Diurnal variations due to perturbation from the Sun's gravity, and Solar weather cycles [36]. Further, if the geometry of the space object we wish to propagate is unknown, then the coefficient of drag, cross-sectional area, and mass are also unavailable.

The fourth acceleration term factors in the presence of other bodies exerting a force on the spacecraft, such as the Sun, Moon, and other planets in the solar system. The summation of accelerations allows for inclusion of any number of perturbing bodies. The acceleration due to the ith perturbing body is

$$\boldsymbol{a}_{\text{ThirdBody}}^{[i]} = \mu_i \left(\frac{\boldsymbol{r}_i}{r_i^3} - \frac{\boldsymbol{r}_{\oplus}^{[i]}}{\left(r_{\oplus}^{[i]}\right)^3} \right) \tag{2.38}$$

where $\boldsymbol{r}_i$ is the position of the spacecraft relative to the ith body and $\boldsymbol{r}_{\oplus}^{[i]}$ is the position of the ith body relative to Earth. This acceleration term can factor in any number of additional gravitational bodies, though for small, distant objects computational accuracy suffers [36]. The final term of Equation 2.35 describes the effect of solar radiation pressure on the spacecraft. Since the light from the sun exerts a small amount of force on any object it encounters, the aggregate effect of the sunlight over an entire orbit can lead to a significant acceleration. The equation for the acceleration caused by the solar radiation pressure is [36]

$$\boldsymbol{a}_{\text{SolarPressure}} = -\frac{p_{\text{srp}} c_R A_{\odot}}{m} \frac{\boldsymbol{r}_{\odot}}{\|\boldsymbol{r}_{\odot}\|} \tag{2.39}$$

where p_{srp} is the force of the solar pressure, $c_R \in [0, 2]$ is the reflectivity of the spacecraft surface, $A_{\odot}$ is the surface area exposed to the sun, and m is the mass of the spacecraft. This effect becomes more significant as the area-to-mass ratio, $\frac{A_{\odot}}{m}$, grows. High area-to-mass objects may deviate significantly from the basic equations of orbital motion due to this effect. Typical spacecraft, however, have an area-to-mass ratio between 0.01 and 0.02 and thus the effect is usually neglected [33].

Of course, the acceleration terms in Equation 2.35 are not a complete list. There are potentially infinite sources of perturbations acting on a spacecraft in orbit, however small. One such example of a perturbation not considered here is the Pioneer anomaly. The Pioneer anomaly is a small, sunward acceleration affecting the Pioneer 10 and 11 spacecraft. The anomaly was first discovered in 1998 through analysis of precise orbit determination results [54]. Later research attributed the mysterious acceleration to asymmetric thermal radiation from the external surface of the spacecraft [55]. For simplicity, we can determine two of the most dominant terms of Equation 2.35: the two-body acceleration and the J_2 non-spherical Earth pertubation. This formulation is easy to implement because it does not require knowledge of the physical properties of the spacecraft. The orbit of a spacecraft affected by only these perturbations is

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a}_{J_2} \tag{2.40}$$

where $\boldsymbol{a}_{J_2}$ is the acceleration to Earth's equatorial bulge. The J_2 perturbation is [36]

$$\begin{aligned}
(\boldsymbol{a}_{J_2})_1 &= -\frac{3}{2}J_2\frac{\mu R_\oplus r_1}{2r^5}\left(1 - \frac{5r_3^2}{r^2}\right)\\
(\boldsymbol{a}_{J_2})_2 &= -\frac{3}{2}J_2\frac{\mu R_\oplus r_2}{2r^5}\left(1 - \frac{5r_3^2}{r^2}\right)\\
(\boldsymbol{a}_{J_2})_3 &= -\frac{3}{2}J_2\frac{\mu R_\oplus r_3}{2r^5}\left(3 - \frac{5r_3^2}{r^2}\right)
\end{aligned} \tag{2.41}$$

where $(\cdot)_i$ indicates the ith component of the vector inside the parenthesis, $R_\oplus$ is the radius of Earth, and

$$J_2 \approx 0.00108263 \tag{2.42}$$

This model is typically sufficient to characterize the motion of large spacecraft and pieces of debris. Now, to use optical observations to estimate the orbit of a spacecraft, we face two challenges: deriving an initial orbit determination (IOD) and continuously updating our knowledge with new measurements. The initial orbit determination problem is well-understood and there are numerous approaches to solving it. For optical systems, we are considering what is called the angles-only orbit determination problem. Since the range from the observer to the object is not available from a single image, the sensor only provides a position on the celestial sphere. Figure 2.15 illustrates this scenario. This figure shows three separate observations over time of a single spacecraft. Since the range is not available, $\boldsymbol{u}_i$ is a unit vector describing the line of sight vector from the sensor to the spacecraft. The spacecraft can theoretically be at any point along this line which extends out to infinity.

The classical angles-only IOD methods attempt to constrain the position and velocity of the object based on the three measured line of sight vectors. We will consider one of the classical techniques: Laplace's Method. Assume that we have three separate observations

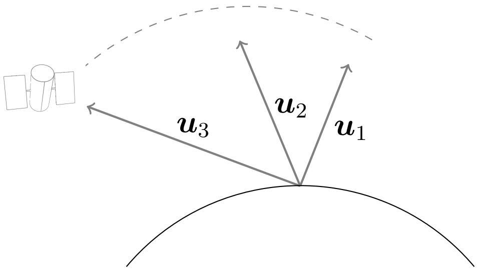

Figure 2.15: Angles-only spacecraft observations.

from a single observer given by the pairs $(t_1, \boldsymbol{u}_1)$, $(t_2, \boldsymbol{u}_2)$, and $(t_3, \boldsymbol{u}_3)$ where t_i is the time corresponding to the ith measurement. Laplace's Method works by interpolating between the observation vectors to create a function $\hat{\boldsymbol{u}}\left(t\right)$. The interpolated observation vector is [36]

$$\hat{\boldsymbol{u}}\left(t\right) = \frac{\left(t-t_2\right)\left(t-t_3\right)}{\left(t_1-t_2\right)\left(t_1-t_3\right)}\boldsymbol{u}_1 + \frac{\left(t-t_1\right)\left(t-t_3\right)}{\left(t_2-t_1\right)\left(t_2-t_3\right)}\boldsymbol{u}_2 + \frac{\left(t-t_1\right)\left(t-t_2\right)}{\left(t_3-t_1\right)\left(t_3-t_2\right)}\boldsymbol{u}_3 \tag{2.43}$$

In order to infer the position and velocity from the expression above, we will also require the derivatives of $\hat{\boldsymbol{u}}\left(t\right)$. The position and velocity vectors of the spacecraft are inferred from these functions. The first derivative of $\hat{\boldsymbol{u}}\left(t\right)$ is

$$\dot{\hat{\boldsymbol{u}}}\left(t\right) = \frac{2t-t_2-t_3}{\left(t_1-t_2\right)\left(t_1-t_3\right)}\boldsymbol{u}_1 + \frac{2t-t_1-t_3}{\left(t_2-t_1\right)\left(t_2-t_3\right)}\boldsymbol{u}_2 + \frac{2t-t_1-t_2}{\left(t_3-t_1\right)\left(t_3-t_2\right)}\boldsymbol{u}_3 \tag{2.44}$$

The second derivative of $\hat{\boldsymbol{u}}\left(t\right)$ is

$$\ddot{\hat{\boldsymbol{u}}}\left(t\right) = \frac{2}{\left(t_1-t_2\right)\left(t_1-t_3\right)}\boldsymbol{u}_1 + \frac{2}{\left(t_2-t_1\right)\left(t_2-t_3\right)}\boldsymbol{u}_2 + \frac{2}{\left(t_3-t_1\right)\left(t_3-t_2\right)}\boldsymbol{u}_3 \tag{2.45}$$

Now, consider the position vector of a spacecraft, described as the position of the spacecraft relative to the sensor plus the vector describing the location of the observer relative to the center of the planet.

$$\boldsymbol{r} = \rho\hat{\boldsymbol{u}} + \boldsymbol{r}_{\text{site}} \tag{2.46}$$

Here, $\rho = \|\boldsymbol{\rho}\|$ is the magnitude of the position of the spacecraft relative to the sensor. We will use this equation to derive a relationship between the observation vector and the true position and velocity of the spacecraft. Taking the first and second derivatives of the equation above yields

$$\begin{aligned} \dot{\boldsymbol{r}} &= \dot{\rho}\hat{\boldsymbol{u}} + \rho\dot{\hat{\boldsymbol{u}}} + \dot{\boldsymbol{r}}_{\text{site}} \\ \ddot{\boldsymbol{r}} &= \ddot{\rho}\hat{\boldsymbol{u}} + 2\dot{\rho}\dot{\hat{\boldsymbol{u}}} + \rho\ddot{\hat{\boldsymbol{u}}} + \ddot{\boldsymbol{r}}_{\text{site}} \end{aligned} \tag{2.47}$$

where the first and second derivatives of the site vector are given by

$$\dot{\boldsymbol{r}}_{\text{site}} = \boldsymbol{\omega}_{\oplus} \times \boldsymbol{r}_{\text{site}}, \qquad \ddot{\boldsymbol{r}}_{\text{site}} = \boldsymbol{\omega}_{\oplus} \times \dot{\boldsymbol{r}}_{\text{site}} \tag{2.48}$$

and $\boldsymbol{\omega}_{\oplus}$ designates the angular rate of Earth. So, Equation 2.47 gives a relationship between an observation vector and the true position and velocity of the spacecraft. Considering only the two-body motion of Equation 2.35, we can write

$$-\frac{\mu}{r^3}\left(\rho\hat{\boldsymbol{u}} + \boldsymbol{r}_{\text{site}}\right) = \ddot{\rho}\hat{\boldsymbol{u}} + 2\dot{\rho}\dot{\hat{\boldsymbol{u}}} + \rho\ddot{\hat{\boldsymbol{u}}} + \ddot{\boldsymbol{r}}_{\text{site}} \tag{2.49}$$

In matrix form, the equation above is

$$\left[\begin{array}{ccc} \hat{\boldsymbol{u}} & 2\dot{\hat{\boldsymbol{u}}} & \ddot{\hat{\boldsymbol{u}}} + \frac{\mu}{r^3}\hat{\boldsymbol{u}} \end{array}\right] \left[\begin{array}{c} \ddot{\rho} \\ \dot{\rho} \\ \rho \end{array}\right] = -\left(\ddot{\boldsymbol{r}}_{\text{site}} + \frac{\mu}{r^3}\boldsymbol{r}_{\text{site}}\right) \tag{2.50}$$

where the first term is a 3 × 3 matrix. This relation, in combination with Equations 2.43 - 2.45, provides a solution for the range, ρ, and the range-rate, $\dot{\rho}$, if the magnitude of $\boldsymbol{r}$ is available. We do not know the magnitude of the spacecraft's position vector, $\boldsymbol{r}$, but we get around this issue by making an initial guess for the magnitude of the radius and iterating until the solution converges [36]. This process provides a position vector, $\boldsymbol{r}$, and a velocity, $\dot{\boldsymbol{r}}$ for the observed spacecraft. There are a number of other approaches to this problem including Gauss's technique, double-r iteration, and Gooding's Method, which are discussed in full in [36]. Any one of these methods is capable of providing an initial estimate from a limited set of observations. With that initial estimate, we can then proceed to an algorithm which will continuously update our estimate as new measurements come in.

Any technique for performing IOD gives the position and velocity of an observed object at only a single point in time from a small set of measurements. Maintaining the orbital knowledge of an object requires a process that continually updates the existing estimate to account for new observations. As always, there are a number of different methods capable of achieving this goal. This section will consider two algorithms: the Extended Kalman Filter and the Unscented Kalman Filter. These will be sufficient for verifying the accuracy of measurements collected in the later chapters.

A basic approach to combining noisy measurements with our current understanding of the state of a system is the Kalman Filter. For a linear system with Gaussian-distributed uncertainties, the Kalman filter provides the minimum-variance (minimum uncertainty) estimate of the state incorporating each measurement and the prior state knowledge. Table 2.2.3 summarizes the process for the discrete Kalman Filter. For each measurement, the filter predicts the state forward from the prior estimate and computes a corrected estimate based on the deviation of the estimate from the measurement and the Kalman Gain.

A key drawback of the Kalman Filter is that it requires linear dynamics and Gaussian uncertainty. Equation 2.40 clearly does not meet the linear requirement, therefore we require another approach. Both methods that we will consider here are approximate versions of the linear Kalman Filter which attempt to pose a nonlinear system in such a way that the original Kalman Filter formulation applies. The first method, the Extended Kalman Filter (EKF),

Table 2.1: Summary of the discrete-time linear Kalman filter [8].

Model	$\boldsymbol{x}_{k+1} = \Phi_k \boldsymbol{x}_k + \Gamma_k \boldsymbol{u}_k + \Upsilon \boldsymbol{w}_k$ $\boldsymbol{z}_k = H_k \boldsymbol{x}_k + \boldsymbol{v}_k$
Prediction	$\hat{\boldsymbol{x}}^-_{k+1} = \Phi \hat{\boldsymbol{x}}^+_k + \Gamma \boldsymbol{u}_k$ $P^-_{k+1} = \Phi_k P^+_k \Phi^T_k + \Gamma_k Q_k \Gamma^T_k$
Kalman Gain	$K_k = P^-_k H^T_k \left(H_k P^-_k H^T_k + R_k \right)^{-1}$
Correction	$\hat{\boldsymbol{x}}^+_k = \hat{\boldsymbol{x}}^-_k + K_k \left(\boldsymbol{z}_k - H_k \hat{\boldsymbol{x}}^-_k \right)$ $P^+_k = \left(I - K_k H_k \right) P^-_k$

attempts to linearize the nonlinear system [8]. Before explaining the procedure, let us first define some variables which will be necessary later on. Define the state vector, $\boldsymbol{x}$ as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \end{bmatrix} \tag{2.51}$$

As in Equation 2.40, $\boldsymbol{r}$ refers to the position vector of a particular object of interest relative to the center of Earth. So the vector $\boldsymbol{x}$ contains both the position and velocity of an object. Now, define the function $F\left(\cdot\right)$, describing the dynamics of the system, as

$$F\left(\boldsymbol{x}\left(t\right), t\right) = \begin{bmatrix} \dot{\boldsymbol{r}} \\ \ddot{\boldsymbol{r}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{r}} \\ -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a}_{J_2} \end{bmatrix} \tag{2.52}$$

In words, $F\left(\boldsymbol{x}\left(t\right), t\right)$ simply contains the differential equations describing the evolution of $\boldsymbol{r}$. Note that here we use the J_2-perturbed model for $\ddot{\boldsymbol{r}}$ from Equation 2.40. Similarly, define a measurement function as

$$\boldsymbol{z}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) \tag{2.53}$$

The measurement function, like the function F, can be nonlinear. Since this dissertation focuses on optical measurements, we will use the measurement model from Equation 2.6. The EKF approach linearizes the dynamics and the measurement model by taking the derivatives of F and h with respect to the state vector. The Jacobian of F at a point $\hat{\boldsymbol{x}}$ is

$$J\left(\hat{\boldsymbol{x}}\left(t\right), t\right) = \left. \frac{\partial F}{\partial \boldsymbol{x}} \right|_{\hat{\boldsymbol{x}}\left(t\right)} \tag{2.54}$$

and the Jacobian of $\boldsymbol{h}$ is

$$H\left(\hat{\boldsymbol{x}}_k\right) = \left. \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \right|_{\hat{\boldsymbol{x}}_k} \tag{2.55}$$

Table 2.2.3 summarizes the Extended Kalman Filter process. Here we consider the continuous-discrete formulation of the EKF. This choice is due to the fact that, while the dynamics are described by continuous differential equations, our sensors return measurements at only discrete points in time. The continuous-discrete formulation allows us to propagate the dynamics in continuous way while the update step is only performed at discrete time intervals. For each measurement, we use the equations in the prediction row to numerically propagate our previous estimate to the current time. That estimate, in combination with the Kalman Gain, provides a corrected estimate that accounts for the measurement uncertainty, $\boldsymbol{v}_k$, and process noise, $\boldsymbol{w}_k$.

Table 2.2: Summary of the continuous-discrete Extended Kalman filter [8].

Model	$\dot{\boldsymbol{x}}\left(t\right) = F\left(\boldsymbol{x}\left(t\right), t\right) + G\left(t\right)\boldsymbol{w}\left(t\right)$ $\boldsymbol{z}_k = \boldsymbol{h}\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k$
Prediction	$\dot{\hat{\boldsymbol{x}}} = F\left(\hat{\boldsymbol{x}}\left(t\right), t\right)$ $\dot{P}\left(t\right) = J\left(\hat{\boldsymbol{x}}\left(t\right), t\right)P\left(t\right) + P\left(t\right)J\left(\hat{\boldsymbol{x}}\left(t\right), t\right)^T + G\left(t\right)Q\left(t\right)G\left(t\right)^T$
Kalman Gain	$K_k = P_k^- H\left(\hat{\boldsymbol{x}}_k^-\right)^T\left[H\left(\hat{\boldsymbol{x}}_k^-\right)P_k^- H\left(\hat{\boldsymbol{x}}_k^-\right) + R_k\right]^{-1}$
Correction	$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K_k\left[\boldsymbol{z}_k - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-\right)\right]$ $P_k^+ = \left[I - K_k H\left(\hat{\boldsymbol{x}}_k^-\right)\right]P_k^-$

An alternative formulation, called the Unscented Kalman Filter, relies on the unscented transform to approximate the covariance, rather than attempting to approximate the equations of motion. The original paper proposing this technique argues that "It is easier to approximate a probability distribution than [...] an arbitrary nonlinear function[56]."

At the heart of the Unscented Kalman Filter is the Unscented Transform, which works to approximate a probability distribution undergoing a nonlinear transformation. The Unscented Transform relies on a choice of a specific set of points in the initial distibution, commonly referred to as sigma points. Figure 2.16 contains an illustration of the unscented transform process. In this simulated scenario, an initial distribution of particles is colored in blue. This distribution is Gaussian, with some mean and covariance. The resulting distribution after propagating the blue particles through a nonlinear function, $F\left(\cdot\right)$ is shown in gray. The sigma points are indicated on the figure with the black "+" marks. The sigma points correspond to the mean of the initial distribution as well as a step along each dimension from the mean. The unscented transform propagates these sigma points through $F\left(\cdot\right)$ and estimates the covariance of the resulting distribution from only this subset of points [56].

This approach is potentially beneficial for the process of orbit determination, because the nonlinear effects can cause significant deviation from the ideal Gaussian of an initial probability distribution. Figure 2.17 contains a simulated example of the effect of the two-body orbital equations of motion on a starting distribution. In this case, the initial position of the spacecraft is a Gaussian random variable with some mean and covariance and the initial velocity is the same for all particles.

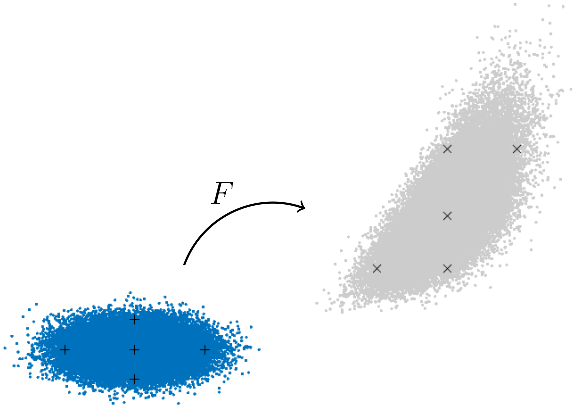

Figure 2.16: Illustration of the unscented transform.

The initial distribution in Figure 2.17 is shown on the left-hand. After propagating the orbit for 2 hours, the resulting distribution is on the right. The particles of the initial distribution have a tendency to "stretch" along the orbital path, curving around the orbit. This effect is due to the fact that the particles closer to or further from Earth are on slightly more eccentric orbits. Relative motion analysis confirms that nearby objects, in the absence of other perturbations, tend to drift primarily in the along-track direction [36].

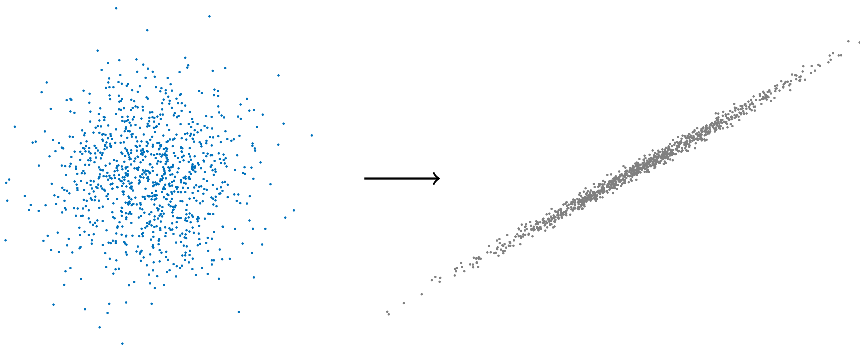

Figure 2.17: Monte Carlo orbital simulation with a Gaussian-distributed initial position.

The Unscented Kalman Filter is often built around an augmented state variable which contains the state, process noise, and measurement noise in a single vector. Though this method is not required, it simplifies the overall formulation. Define the augmented state vector at time step k as $\boldsymbol{x}_k^a$ where

$$\boldsymbol{x}_k^a = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{w}_k \\ \boldsymbol{v}_k \end{bmatrix} \tag{2.56}$$

where $\boldsymbol{w}_k \sim N\left(0, Q\left(t\right)\right)$ is the process noise and $\boldsymbol{v}_k \sim N\left(0, R\left(t\right)\right)$ is the measurement noise. The covariance corresponding to $\boldsymbol{x}_k^a$ at time step k is

$$P_k^a = \begin{bmatrix} P_k^+ & P_k^{xw} & P_k^{xv} \\ \left(P_k^{xw}\right)^T & Q_k & P_k^{wv} \\ \left(P_k^{xv}\right)^T & \left(P_k^{wv}\right)^T & R_k \end{bmatrix} \tag{2.57}$$

where P_k^{xw} is the correlation between the state error and process noise, P_k^{xv} is the correlation between the state error and the measurement noise, and P_k^{wv} is the correlation between the process noise and measurement noise. Conveniently, P_k^{xw}, P_k^{xv}, and P_k^{wv} are zero for most systems and need not be considered [8]. Augmenting the state variable yields a simple formulation for the choice of sigma points. Define χ_i for $i = 0, \ldots, 2L$ as the set of sigma points from the initial probability distribution which will approximate the resulting distribution. The standard definition of χ_i is [56]

$$\begin{aligned} \chi_0 &= \hat{\boldsymbol{x}}_k^a \\ \chi_i &= \hat{\boldsymbol{x}}_k^a + \left(\sqrt{\left(L+\lambda\right) P_k^a}\right)_i, \qquad i = 1, \ldots, L \\ \chi_i &= \hat{\boldsymbol{x}}_k^a - \left(\sqrt{\left(L+\lambda\right) P_k^a}\right)_i, \qquad i = L+1, \ldots, 2L \end{aligned} \tag{2.58}$$

where $\left(\cdot\right)_i$ denotes the ith column of the expression inside the parenthesis. The term L is the length of the vector $\boldsymbol{x}_k^a$. The parameter λ is called the composite scaling parameter, which is typically defined by [8]

$$\lambda = \alpha^2 \left(L + \kappa\right) - L \tag{2.59}$$

where α gives the spread of the sigma points and is typically small, and κ, for non-scalar systems, is set to [8]

$$\kappa = 3 - L \tag{2.60}$$

A set of weightings provide the mean and covariance estimates from the complete set of sigma points. It is typical to use two sets of weightings: one for estimation of the mean and one for the covariance matrices. The standard choice of weightings is shown below [8].

$$\begin{aligned} W_0^m &= \frac{\lambda}{L+\lambda} \\ W_0^c &= W_0^m + \left(1 - \alpha^2 + \beta\right) \\ W_i^m &= W_i^c = \frac{1}{2\left(L+\lambda\right)}, \qquad i = 1, \ldots, 2L \end{aligned} \tag{2.61}$$

Here β is another tuning parameter; typically $\beta = 2$ is sufficient [8]. The term W_i^m denotes the weightings required to compute the mean from the set of sigma points and W_i^c contains the weightings used to compute the covariance terms. Table 2.2.3 summarizes the process for the Unscented Kalman Filter.

In words, the UKF begins by defining the sigma points based on the current covariance P_k. Then, the UKF propagates those points and estimates the mean of the resulting distribution as well as the necessary covariances. The Kalman gain uses these results to update the state in the same way as the standard Kalman Filter.

Table 2.3: Summary of the Unscented Kalman filter [8].

Model	$\chi_{i,k+1} = F\left(\chi_{i,k}\right)$ $\boldsymbol{\gamma}_k = \boldsymbol{h}\left(\chi_k\right)$
Prediction	$\hat{\boldsymbol{x}}_k^- = \sum_{i=0}^{2L} W_i^m \chi_{i,k}$ $P_k^- = \sum_{i=0}^{2L} W_i^c \left[\chi_{i,k}^x - \hat{\boldsymbol{x}}_k^-\right] \left[\chi_{i,k}^x - \hat{\boldsymbol{x}}_k^-\right]^T$ $P_k^{xz} = \sum_{i=0}^{2L} W_i^c \left[\chi_{i,k}^x - \hat{\boldsymbol{x}}_k^-\right] \left[\gamma_k - \hat{\boldsymbol{z}}_k^-\right]^T$ $P_k^{zz} = \sum_{i=0}^{2L} W_i^c \left[\gamma_k - \hat{\boldsymbol{z}}_k^-\right] \left[\gamma_k - \hat{\boldsymbol{z}}_k^-\right]^T$
Kalman Gain	$K_k = P_k^{xz} \left(P_k^{vv}\right)^{-1}$
Correction	$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K_k \left[z_k - \boldsymbol{h}\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{v}_k\right)\right]$ $P_k^+ = P_k^- - K_k P_k^{vv} K_k^T$

The algorithms for orbit estimation described in this section are the end goal of all of the work in this dissertation. The methods for image analysis derived in the coming sections are intended to augment our ability to extract space objects from unresolved imagery. The goal of that type of data analysis is, ultimately, to report the angles-only measurement data to an orbit determination routine that is capable of maintaining a complete catalog of space objects in orbit around Earth. So, even though the remaining Chapters focus on the sensor-level data processing pipeline, these orbit estimation algorithms would play a key role in an operational system.

Chapter 3

Dynamic and Geometric Analysis of the Optical Observer

This chapter will focus on deriving dynamic equations of motion for space objects viewed through an optical sensor. Chapter 2 touched on the fact that typical data reduction techniques for SSA rely on broad assumptions about the motion of a particular sensor. Though these approaches are capable of generating valuable data, they limit the set of useful sensors to only those which are tasked in a certain manner. In an effort to expand the number of contributing sensors, we may instead work to consider as general a motion model as possible. This approach will allow for a greater quantity and quality of data from ground- and space-based observers. Here we will work to derive equations of motion for an object viewed through an optical sensor. These equations of motion will rely on as few assumptions about the motion of the sensor as possible to remain applicable to a general population of optical observers.

This chapter is separated into four sections. The first explores motion of space objects across the image plane of a sensor adhering to the pinhole camera model. Each sub-section considers the equations of motion arising from a certain set of assumptions constraining the sensor and space object's motion. The second section considers the motion in a distortion-corrected image plane. In this case, a correct polynomial like the one discussed in the previous chapter modifies an image to produce an "un-distorted" result. Here, we will define the un-distorted model and derive a set of equations of motion arising from that definition. The remaining two sections of this chapter focus on geometric approaches to understanding image data. Section 3.3 considers a geometric representation of the motion of intertially-fixed objects across the image plane. Section 3.4 extends our analysis to include the possibility of multiple sensors operating in concert.

3.1 Motion in the Pinhole Image Plane

Figure 3.1 illustrates a cross-section of the geometry between an optical sensor and a spacecraft. The relative position between the sensor and the spacecraft is noted by the term $\boldsymbol{\rho}$, which is expressed in a body-fixed frame of the sensor where the sensor's pointing direction is always aligned with the $\hat{e}_3$ unit vector.

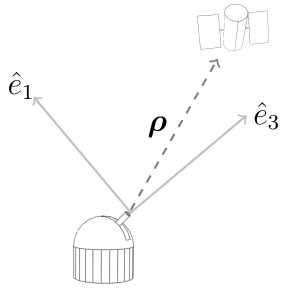

Figure 3.1: A spacecraft's relative position in the sensor's body-fixed coordinate system.

There are a number of different ways to frame this problem. First, it is necessary to derive a relationship between the dynamics of a signal in the image plane and the dynamics of the corresponding object's relative position vector, $\boldsymbol{\rho}$. Recall the projection from $\mathbb{R}^3$ to $\mathbb{R}^2$ for an optical sensor according to the pinhole camera model. The collinearity equation is

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\frac{f}{\rho_3}\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{3.1}$$

This equation provides a relation between the image-plane coordinates, (x, y), and the relative position of an object of interest to the sensor, $\boldsymbol{\rho}$, so it will serve as a starting point for deriving dynamic relationships between the two. Of course, this is a simple model which does not capture all of the nuance of a real-world sensor. Theoretical results from this model can provide valuable insight into the fundamental behavior of optical sensors, however. Current research in the area of optical navigation in the Earth-Moon system leverages this model for theoretical analysis of the shape of the Moon in an image [13, 57, 14].

In general, a direct solution for $\boldsymbol{\rho}$ as a function of time may not be available, so Equation 3.1 alone is insufficient. The derivative of the Collinearity equation, however, will provide some insight into the connection between the image-plane dynamics and the evolution of $\boldsymbol{\rho}$. We will begin by taking the time derivatives of the x and y components of the Collinearity equation. The derivative of the x coordinate is

$$\dot{x} = -f\frac{\dot{\rho}_1}{\rho_3} + f\frac{\rho_1\dot{\rho}_3}{\rho_3^2} \tag{3.2}$$

Equation 3.1 shows that $x = -f\frac{\rho_1}{\rho_3}$, so the equation above is equivalent to

$$\dot{x} = -f\frac{\dot{\rho}_1}{\rho_3} - x\frac{\dot{\rho}_3}{\rho_3} \tag{3.3}$$

The first derivative of the y coordinate with respect to time is

$$\dot{y} = -f\frac{\dot{\rho}_2}{\rho_3} + f\frac{\rho_2\dot{\rho}_3}{\rho_3^2} \tag{3.4}$$

Again, Equation 3.1 shows that $y = -f\frac{\rho_2}{\rho_3}$, so this equation becomes

$$\dot{y} = -f\frac{\dot{\rho}_2}{\rho_3} - y\frac{\dot{\rho}_3}{\rho_3} \tag{3.5}$$

In some situations even the first derivative with respect to time may not be available. This will be the case for nearly all orbiting space objects because of the dynamics in Equation 2.40 so a higher-order relation is necessary. The second derivative of the x coordinate with respect to time is

$$\ddot{x} = 2f\frac{\dot{\rho}_1\dot{\rho}_3}{\rho_3^2} - 2f\frac{\rho_1\dot{\rho}_3^2}{\rho_3^3} - f\frac{\ddot{\rho}_1}{\rho_3} + f\frac{\rho_1\ddot{\rho}_3}{\rho_3^2} \tag{3.6}$$

Equation 3.1 again allows for some simplification

$$\ddot{x} = 2f\frac{\dot{\rho}_1\dot{\rho}_3}{\rho_3^2} + 2x\frac{\dot{\rho}_3^2}{\rho_3^2} - f\frac{\ddot{\rho}_1}{\rho_3} - x\frac{\ddot{\rho}_3}{\rho_3} \tag{3.7}$$

Further, by writing the equation above as

$$\ddot{x} = 2\frac{\dot{\rho}_3}{\rho_3}\left(f\frac{\dot{\rho}_1}{\rho_3} + x\frac{\dot{\rho}_3}{\rho_3}\right) - f\frac{\ddot{\rho}_1}{\rho_3} - x\frac{\ddot{\rho}_3}{\rho_3} \tag{3.8}$$

we see that the term in parentheses is equivalent to $-\dot{x}$ according to Equation 3.3.

$$\ddot{x} = -2\dot{x}\frac{\dot{\rho}_3}{\rho_3} - x\frac{\ddot{\rho}_3}{\rho_3} - f\frac{\ddot{\rho}_1}{\rho_3} \tag{3.9}$$

Similarly, the second derivative of the y coordinate with respect to time is

$$\ddot{y} = 2f\frac{\dot{\rho}_2\dot{\rho}_3}{\rho_3^2} - 2f\frac{\rho_2\dot{\rho}_3^2}{\rho_3^3} - f\frac{\ddot{\rho}_2}{\rho_3} + f\frac{\rho_2\ddot{\rho}_3}{\rho_3^2} \tag{3.10}$$

The Collinearity equation shows that

$$\ddot{y} = 2f\frac{\dot{\rho}_2\dot{\rho}_3}{\rho_3^2} + 2y\frac{\dot{\rho}_3^2}{\rho_3^2} - f\frac{\ddot{\rho}_2}{\rho_3} - y\frac{\ddot{\rho}_3}{\rho_3} \tag{3.11}$$

Similar to the equation for $\ddot{x}$, we can write

$$\ddot{y} = 2\frac{\dot{\rho}_3}{\rho_3}\left(f\frac{\dot{\rho}_2}{\rho_3} + 2y\frac{\dot{\rho}_3}{\rho_3}\right) - f\frac{\ddot{\rho}_2}{\rho_3} - y\frac{\ddot{\rho}_3}{\rho_3} \tag{3.12}$$

Finally, substitution of Equation 3.5 for the term in the parentheses gives

$$\ddot{y} = -2\dot{y}\frac{\dot{\rho}_3}{\rho_3} - y\frac{\ddot{\rho}_3}{\rho_3} - f\frac{\ddot{\rho}_2}{\rho_3} \tag{3.13}$$

So, in summary, the first derivatives of the Collinearity Equation are

$$\begin{aligned} \dot{x} &= -f\frac{\dot{\rho}_1}{\rho_3} - x\frac{\dot{\rho}_3}{\rho_3} \\ \dot{y} &= -f\frac{\dot{\rho}_2}{\rho_3} - y\frac{\dot{\rho}_3}{\rho_3} \end{aligned} \tag{3.14}$$

and the second derivatives are

$$\begin{aligned} \ddot{x} &= -2\dot{x}\frac{\dot{\rho}_3}{\rho_3} - x\frac{\ddot{\rho}_3}{\rho_3} - f\frac{\ddot{\rho}_1}{\rho_3} \\ \ddot{y} &= -2\dot{y}\frac{\dot{\rho}_3}{\rho_3} - y\frac{\ddot{\rho}_3}{\rho_3} - f\frac{\ddot{\rho}_2}{\rho_3} \end{aligned} \tag{3.15}$$

These dynamic equations relate x and y to the relative dynamics of an observed object. By assuming some dynamics for an object and inserting them into the ρ term of Equations 3.14 and Equations 3.15 we can derive a relation for that object's motion across the image plane.

3.1.1 Constant Rotation Observer, Inertially-Fixed Object

First, consider the motion of stars in the body-fixed coordinate frame of the sensor. The motion of a star in angle space across the celestial sphere is typically measured on the order of mili-arcseconds per year [6]. One mili-arcsecond per year is approximately 30 pico-arcesconds per second. So, based on this knowledge, we can safely assume stars to be inertially fixed over the integration time on the order of seconds for an optical sensor. Thus, a star's motion across the image plane is determined entirely by the angular rate of the sensor. In order to proceed further we must make some assumptions about the motion of the sensor. Consider an optical sensor which maintains a constant angular rate over its integration time. If the relative position of a star is given by $\boldsymbol{\rho}$, then the relative position vector evolves according to [58]

$$\dot{\boldsymbol{\rho}} = -\boldsymbol{\omega} \times \boldsymbol{\rho} \tag{3.16}$$

which is equivalently,

$$\dot{\boldsymbol{\rho}} = -\tilde{\boldsymbol{\omega}}\boldsymbol{\rho} \tag{3.17}$$

where $\tilde{\boldsymbol{\omega}}$ is the skew-symmetric matrix

$$\tilde{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{3.18}$$

So, expanding Equation 3.17, we have

$$\begin{aligned}\dot{\rho}_1 &= \omega_3\rho_2 - \omega_2\rho_3\\ \dot{\rho}_2 &= -\omega_3\rho_1 + \omega_1\rho_3\\ \dot{\rho}_3 &= \omega_2\rho_1 - \omega_1\rho_2\end{aligned} \tag{3.19}$$

These dynamic equations for the three components of the relative position vector will allow for a connection between the assumed dynamics in Equation 3.17 and the image-plane dynamics in Equation 3.14. Insertion of the relations for $\dot{\rho}_1$ and $\dot{\rho}_3$ into the $\dot{x}$ equation yields

$$\dot{x} = -f\frac{\omega_3\rho_2 - \omega_2\rho_3}{\rho_3} - x\frac{\omega_2\rho_1 - \omega_1\rho_2}{\rho_3} \tag{3.20}$$

Expanding each term, this equation becomes

$$\dot{x} = -f\frac{\omega_3\rho_2}{\rho_3} + f\frac{\omega_2\rho_3}{\rho_3} - x\frac{\omega_2\rho_1}{\rho_3} + x\frac{\omega_1\rho_2}{\rho_3} \tag{3.21}$$

Finally, substituting components of the Collinearity equation provides an equation expressed entirely in terms of the image plane coordinates, x and y.

$$\dot{x} = \omega_3 y + f\omega_2 + \frac{1}{f}\omega_2 x^2 - \frac{1}{f}\omega_1 xy \tag{3.22}$$

The result above is important because it rigorously shows that the full position in $\mathbb{R}^3$ of an inertially-fixed object is not relevant to the resulting image. Rather, an object's motion across the image plane of a rotating observer is fully defined by its initial (x, y) projection into the focal plane and the angular rate of the sensor. A similar derivation follows for the y equation. Inserting the relations for $\dot{\rho}_2$ and $\dot{\rho}_3$ into the relation for $\dot{y}$ in Equation 3.14, we find that

$$\dot{y} = -f\frac{-\omega_3\rho_1 + \omega_1\rho_3}{\rho_3} - y\frac{\omega_2\rho_1 - \omega_1\rho_2}{\rho_3} \tag{3.23}$$

As before, we expand each of the terms to write

$$\dot{y} = f\frac{\omega_3\rho_1}{\rho_3} - f\frac{\omega_1\rho_3}{\rho_3} - y\frac{\omega_2\rho_1}{\rho_3} + y\frac{\omega_1\rho_2}{\rho_3} \tag{3.24}$$

Finally, consideration of the Collinearity Equation produces an expression which contains no terms of the relative position vector.

$$\dot{y} = -\omega_3 x - f\omega_1 + \frac{1}{f}\omega_2 xy - \frac{1}{f}\omega_1 y^2 \tag{3.25}$$

These two differential equations allow for propagation of an inertially-fixed object across the image plane of a rotating sensor without need for knowledge of the true location of that object in $\mathbb{R}^3$. Direct solution of these equations, however, is complicated.

3.1.2 Single-Axis Constant Rotation Observer, Inertially-Fixed Object

Some additional assumptions will allow for a closed-form solution to Equation 3.22 and 3.25. Consider an observer rotating about a single axis. If the axis of rotation is aligned with the body-fixed y axis, then $\omega_3 = \omega_2 = 0$. Equations 3.22 and 3.25 are

$$\begin{aligned} \dot{x} &= -\frac{1}{f}\omega_1 xy \\ \dot{y} &= -f\omega_1 - \frac{1}{f}\omega_1 y^2 \end{aligned} \tag{3.26}$$

First, consider the equation for $\dot{y}$, since it has no coupling between the x and y terms. We have

$$\dot{y} = -f\omega_1 - \frac{1}{f}\omega_1 y^2 \tag{3.27}$$

which is a separable ODE. The equation above is equivalent to

$$\frac{dy}{dt} = -\frac{\omega_1}{f}\left(f^2 + y^2\right) \tag{3.28}$$

which may be rearranged to find

$$\frac{dy}{f^2 + y^2} = -\frac{\omega_1}{f}dt \tag{3.29}$$

Note that since f is real and positive, then $f^2 + y^2 \neq 0$ for any $y \in \mathbb{R}$ so dividing by $(f^2 + y^2)$ does not introduce any singularities. Now, integrate both sides of this equation.

$$\int \frac{dy}{f^2 + y^2} = -\int \frac{\omega_1}{f}dt \tag{3.30}$$

Factoring $\frac{1}{f^2}$ from the left-hand side gives

$$\frac{1}{f^2}\int \frac{dy}{1 + y^2/f^2} = -\int \frac{\omega_1}{f}dt \tag{3.31}$$

and evaluation of the integral yields

$$\frac{1}{f}\tan^{-1}\left(\frac{y}{f}\right) = -\frac{\omega_1}{f}t + c_1 \tag{3.32}$$

where c_1 is an unknown constant. When solved for y, the equation above becomes

$$y\left(t\right) = -f\tan\left(\omega_1 t - c_1 f\right) \tag{3.33}$$

From the initial condition, $y\left(t_0\right)$, we can find the constant c_1.

$$y\left(t_0\right)=-f\tan\left(\omega_1 t_0-c_1 f\right) \quad\implies\quad c_1=\frac{1}{f}\left(\omega_1 t_0-\tan^{-1}\left(-\frac{y\left(t_0\right)}{f}\right)\right) \tag{3.34}$$

Insertion of c_1 back into the original equation yields

$$y\left(t\right)=-f\tan\left(\omega_1\left(t-t_0\right)-\tan^{-1}\left(\frac{y\left(t_0\right)}{f}\right)\right) \tag{3.35}$$

To simplify the equation above, we can write

$$y\left(t\right)=-f\tan\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right) \tag{3.36}$$

where the angle ϕ is

$$\phi\left(t\right)=\tan^{-1}\left(-\frac{y\left(t\right)}{f}\right) \tag{3.37}$$

For now, we will simply use this definition for convenience. Later on its relation to the geometry of the problem will become apparent. Now, the $\dot{x}$ equation is

$$\dot{x}=-\frac{1}{f}\omega_1 x\left(-f\tan\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right)\right) \tag{3.38}$$

We can write

$$\frac{dx}{dt}=\omega_1 x\tan\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right) \tag{3.39}$$

and integrate both sides to solve for an explicit equation for $x\left(t\right)$.

$$\int\frac{dx}{x}=\int\omega_1\tan\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right)dt \tag{3.40}$$

The integral evaluates to

$$\log\left(x\left(t\right)\right)=\omega_1\left(-\frac{\log\left(\cos\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right)\right)}{\omega_1}\right)+c_2=-\log\left(\cos\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right)\right)+c_2 \tag{3.41}$$

where c_2 is an unknown constant. Now, take the exponential of both sides of this equation

$$\exp\left(\log x\left(t\right)\right)=\exp\left(-\log\left(\cos\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right)\right)+c_2\right) \tag{3.42}$$

Evaluation of the left- and right-hand sides of this equation produces

$$x\left(t\right)=\exp\left(c_2\right)\sec\left(\omega_1\left(t-t_0\right)+\phi\left(t_0\right)\right) \tag{3.43}$$

To simplify notation, define a third constant, $c_3=\exp\left(c_2\right)$. Now, the initial condition $x\left(t_0\right)$ provides a solution for the constant c_3.

$$x\left(t_0\right)=c_3\sec\left(\phi\left(t_0\right)\right) \quad\implies\quad c_3=\frac{x\left(t_0\right)}{\sec\left(\phi\left(t_0\right)\right)}=x\left(t_0\right)\cos\left(\phi\left(t_0\right)\right) \tag{3.44}$$

Finally, inserting c_3 into the solution for $x(t)$ yields

$$x(t) = \frac{x(t_0)\cos(\phi(t_0))}{\cos(\omega_1(t-t_0)+\phi(t_0))} \tag{3.45}$$

Thus we have found an analytical solution for the motion of an inertially-fixed object across the image plane of a sensor rotating about a single axis. So, in summary, for the case where $\omega_2 = \omega_3 = 0$,

$$\begin{aligned} x(t) &= \frac{x(t_0)\cos(\phi(t_0))}{\cos(\omega_1(t-t_0)+\phi(t_0))} \\ y(t) &= -f\tan(\omega_1(t-t_0)+\phi(t_0)) \end{aligned} \tag{3.46}$$

where the angle ϕ is

$$\phi(t) = \tan^{-1}\left(-\frac{y(t)}{f}\right) \tag{3.47}$$

A similar derivation exists for the case where $\omega_1 = \omega_3 = 0$ and $\omega_2 \neq 0$. Under this set of assumptions, Equations 3.22 and 3.25 become

$$\begin{aligned} \dot{x} &= f\omega_2 + \frac{1}{f}\omega_2 x^2 \\ \dot{y} &= \frac{1}{f}\omega_2 xy \end{aligned} \tag{3.48}$$

The solutions for the differential equations above are

$$\begin{aligned} x(t) &= f\tan(\omega_2(t-t_0)+\theta(t_0)) \\ y(t) &= \frac{y(t_0)\cos(\theta(t_0))}{\cos(\omega_2(t-t_0)+\theta(t_0))} \end{aligned} \tag{3.49}$$

where the angle θ is

$$\theta(t) = \tan^{-1}\left(-\frac{x(t)}{f}\right) \tag{3.50}$$

3.1.3 Constant Rotation Observer Approximate Solutions

To derive a closed-form solution for an arbitrary $\boldsymbol{\omega}$ we require a different set of assumptions that will sufficiently simplify the differential equations for $\dot{x}$ and $\dot{y}$. Specifically, we will look at the higher-order terms of Equations 3.22 and 3.25. If these terms could be approximated as zero, then the two differential equations will be linear in x and y.

Consider the geometry illustrated in Figure 3.3. In this figure, the pointing direction of the sensor is indicated by the gray line orthogonal to the focal plane. The variable θ is a measure

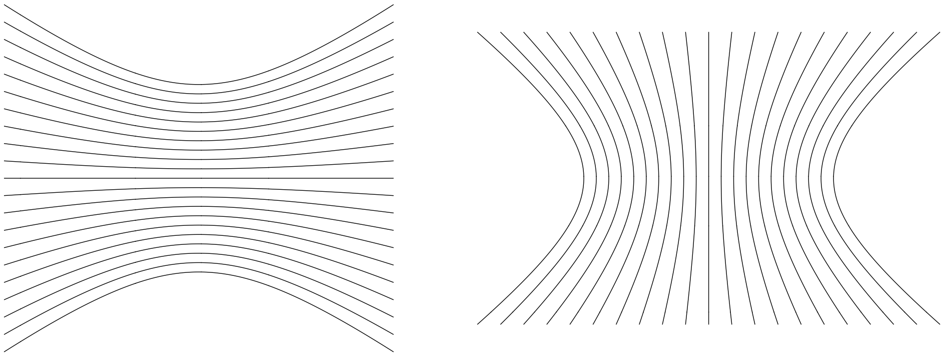

Figure 3.2: Illustrations of star paths across the focal plane from the point of view of a sensor rotating about the body-fixed y axis (left) and the body-fixed x axis (right).

of the angle between the pointing vector and the line of sight vector for a particular object in view of the sensor. The relationship between θ, Δx, and the focal length, f is

$$\tan(\theta) = \frac{x}{f} \tag{3.51}$$

From this equation and Figure 3.3 it is apparent that, as θ approaches zero, Δx also approaches zero. Since f is constant for a particular sensor, this implies that as θ approaches zero, the ratio between Δx and f approaches zero as well.

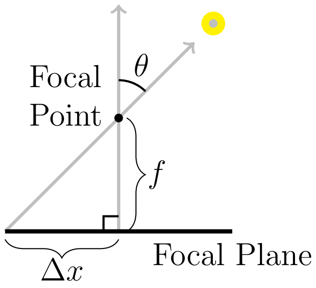

Figure 3.3: Angle between the sensor pointing vector and a star signal's line of sight vector.

Consider one of the higher-order terms from Equation 3.22. We can write

$$\frac{x^2}{f} = f\left(\frac{x}{f}\right)^2 = f\tan^2(\theta) \tag{3.52}$$

So, for a particular field of view and focal length, the equation above defines the maximum magnitude of the higher-order term $\frac{x^2}{f}$. If we assume Δx in Figure 3.3 to correspond to an object exactly on the edge of the field of view of the sensor, then θ is equal to the field of view. Figure 3.4 visualizes the maximum possible values of $\frac{x^2}{f}$ for three focal lengths and a

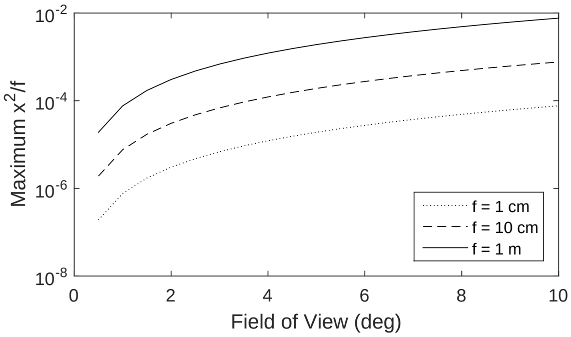

Figure 3.4: Sample values for $\frac{x^2}{f}$ versus field of view size.

range of fields of view. Of course, some of these paramters lead to unrealistically small or large CCD sensors but the data illustrates the theoretical behavior.

With the small field of view assumption that $x \ll f$ and $y \ll f$, we find that the higher-order terms are typically small. Thus, we may write Equations 3.22 and 3.25 as

$$\dot{x} \approx f\omega_3 y + f\omega_2 \tag{3.53}$$

and

$$\dot{y} \approx -\omega_3 x - f\omega_1 \tag{3.54}$$

which are linear, first-order ODEs. Combining these two equations in matrix form yields

$$\left[\begin{array}{c} \dot{x} \\ \dot{y} \end{array}\right] = \left[\begin{array}{cc} 0 & \omega_3 \\ -\omega_3 & 0 \end{array}\right] \left[\begin{array}{c} x \\ y \end{array}\right] + f \left[\begin{array}{c} \omega_2 \\ -\omega_1 \end{array}\right] \tag{3.55}$$

Since this system of equations is linear, we can solve it directly for a closed-form solution of $x(t)$ and $y(t)$. For a linear ODE, the homogeneous solution is

$$\left[\begin{array}{c} x_h(t) \\ y_h(t) \end{array}\right] = \exp\left(\left[\begin{array}{cc} 0 & \omega_3 \\ -\omega_3 & 0 \end{array}\right] t\right) = \left[\begin{array}{cc} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{array}\right] \tag{3.56}$$

Note that the result here is equivalent to a planar rotation about the origin with a magnitude equal to $\omega_3 t$. Now, the Variation of Constants formula gives

$$\begin{aligned} \left[\begin{array}{c} x(t) \\ y(t) \end{array}\right] = & \left[\begin{array}{cc} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{array}\right] \left[\begin{array}{c} x(t_0) \\ y(t_0) \end{array}\right] \\ & + \int_0^t \exp\left(\left[\begin{array}{cc} 0 & \omega_3 \\ -\omega_3 & 0 \end{array}\right](t-s)\right) f \left[\begin{array}{c} \omega_2 \\ -\omega_1 \end{array}\right] ds \end{aligned} \tag{3.57}$$

where $x(t_0)$ and $y(t_0)$ denote the initial conditions. Under the assumption that the angular rate, $\boldsymbol{\omega}$, and the focal length, f, are constants, we pull these terms outside of the integral on the right-hand side.

$$\begin{aligned}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} \\ &+ f \left\{ \int_0^t \exp\left(\begin{bmatrix} 0 & \omega_3(t-s) \\ -\omega_3(t-s) & 0 \end{bmatrix} \right) ds \right\} \begin{bmatrix} \omega_2 \\ -\omega_1 \end{bmatrix}\end{aligned} \tag{3.58}$$

Finally, the solution of the integral gives

$$\begin{aligned}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} \\ &+ \frac{f}{\omega_3} \begin{bmatrix} \sin(\omega_3 t) & -\cos(\omega_3 t) + 1 \\ \cos(\omega_3 t) - 1 & \sin(\omega_3 t) \end{bmatrix} \begin{bmatrix} \omega_2 \\ -\omega_1 \end{bmatrix}\end{aligned} \tag{3.59}$$

Rearranging the second term creates a slightly more convenient form, which is

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} + \frac{f}{\omega_3} \begin{bmatrix} \cos(\omega_3 t) - 1 & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) - 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{3.60}$$

Note that the above relation is undefined for $\omega_3 = 0$, or when the rotation rate about the boresight of the sensor is zero. Looking back at Equation 3.55, we see that, if $\omega_3 = 0$, then

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = f \begin{bmatrix} \omega_2 \\ -\omega_1 \end{bmatrix} \tag{3.61}$$

for which the solution is simply

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} + f \begin{bmatrix} \omega_2 \\ -\omega_1 \end{bmatrix} t \tag{3.62}$$

Now let us consider the behavior of the system in Equation 3.60 as ω_3 approaches zero. The limit of the first term is simply a zero-magnitude rotation of the initial conditions about the origin

$$\lim_{\omega_3 \to 0} \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \tag{3.63}$$

The second term in Equation 3.60 contains the undefined portion of interest. L'Hopital's rule shows that

$$\begin{aligned}\lim_{\omega_3 \to 0} \frac{f}{\omega_3} \begin{bmatrix} \cos(\omega_3 t) - 1 & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) - 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} &= \lim_{\omega_3 \to 0} f \begin{bmatrix} -\sin(\omega_3 t) & \cos(\omega_3 t) \\ -\cos(\omega_3 t) & -\sin(\omega_3 t) \end{bmatrix} t \\ &= f \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} t\end{aligned} \tag{3.64}$$

Inserting the two results above into Equation 3.60 yields

$$\begin{aligned}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} + f \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} t \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \\ &= \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} + f \begin{bmatrix} \omega_2 \\ -\omega_1 \end{bmatrix} t \end{aligned} \tag{3.65}$$

which is equivalent to Equation 3.62. Thus we have shown that as ω_3 approaches zero, the system in Equation 3.60 approaches that in Equation 3.62. Though the general solution is undefined at $\omega_3 = 0$, we can correct this problem by defining the solution piecewise. If ω_3 is ever identically zero, then the model in Equation 3.62 is necessary. Otherwise 3.60 completely captures the behavior of the original ODE.

Now, there is another way of examining Equation 3.60 that provides additional insight into the behavior of the overall system. We have already seen that the Equation describes a planar rotation, but about which point is this rotation centered? To determine this point, which we will refer to as the "center of rotation" in the image, we will look at the second term in Equation 3.60.

$$\frac{f}{\omega_3} \begin{bmatrix} \cos(\omega_3 t) - 1 & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) - 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{3.66}$$

Note that there is a similar structure to that of the Collinearity Equation. Define the center of rotation, (x_c, y_c), as the projection of the angular rate vector into the image plane

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = -f \frac{1}{\omega_3} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{3.67}$$

If we write the angular rate vector as the magnitude of the angular rate multiplied by an axis of rotation,

$$\boldsymbol{\omega} = \|\boldsymbol{\omega}\| \boldsymbol{a} \tag{3.68}$$

then Equation 3.67 becomes

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = -f \frac{1}{\|\boldsymbol{\omega}\| a_3} \begin{bmatrix} \|\boldsymbol{\omega}\| a_1 \\ \|\boldsymbol{\omega}\| a_2 \end{bmatrix} = -f \frac{1}{a_3} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{3.69}$$

which describes the projection of the axis of rotation into the image plane. Now insert this definition for the center of rotation into Equation 3.60.

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} - \begin{bmatrix} \cos(\omega_3 t) - 1 & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) - 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{3.70}$$

The coefficient on the center of rotation vector is a rotation matrix minus the identity matrix.

$$\begin{aligned}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = & \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} \\ & - \left(\begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x_c \\ y_c \end{bmatrix} \end{aligned} \tag{3.71}$$

Expansion of the second term yields

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} - \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{3.72}$$

Then, combining like terms, we have

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \left(\begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix} \right) \tag{3.73}$$

Consider a new coordinate system with its origin located at the center of rotation. Define $(\hat{x}, \hat{y})$ as a point in this new coordinate system. The new coordinates are related to the old coordinates by

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{3.74}$$

Inserting this change of variables leads to

$$\begin{bmatrix} \hat{x}(t) \\ \hat{y}(t) \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} \hat{x}(t_0) \\ \hat{y}(t_0) \end{bmatrix} \tag{3.75}$$

which describes a pure rotation about the origin of the $(\hat{x}, \hat{y})$ coordinate system. Thus we have shown that, according to the linearized model of star motion, all stars in an image rotate with a constant rate about the point where the axis of rotation intersects the image plane. The fact that stars trace circles around the center of rotation, as defined in Equation 3.67, is a key result that will be leveraged in the later chapters.

3.1.4 Constant Rotation Observer, Constant Velocity Object

Up until now we have only considered objects which appear to be inertially-fixed. This assumption only works for very distant space objects and thus may not be appropriate for describing the motion of an object orbiting Earth. Now we will examine the case where a sensor is viewing an object of interest that has some motion relative to the sensor. For now, we will approximate this motion as constant. In general, the accelerations in Equation 2.40 are small on the time scale of the exposure time of a single image, so this description should capture the expected behavior of an orbiting object. Considering a space object's motion relative to the sensor, its relative position vector evolves according to [58]

$$\dot{\boldsymbol{\rho}} = \dot{\boldsymbol{\rho}}_{nr} - \boldsymbol{\omega} \times \boldsymbol{\rho} \tag{3.76}$$

where $\dot{\boldsymbol{\rho}}_{nr}$ is the velocity in the inertial frame of a particle relative to the sensor. This velocity term is assumed to be constant. The individual components of $\dot{\boldsymbol{\rho}}_{nr}$ are

$$\begin{aligned} \dot{\rho}_1 &= \left(\dot{\rho}_{nr}\right)_1 + \omega_3 \rho_2 - \omega_2 \rho_3 \\ \dot{\rho}_2 &= \left(\dot{\rho}_{nr}\right)_2 - \omega_3 \rho_1 + \omega_1 \rho_3 \\ \dot{\rho}_3 &= \left(\dot{\rho}_{nr}\right)_3 + \omega_2 \rho_1 - \omega_1 \rho_2 \end{aligned} \tag{3.77}$$

Here, $(\dot{\rho}_{nr})_1$ for $i = 1, 2, 3$ is the ith element of $\dot{\boldsymbol{\rho}}_{nr}$. We will proceed in the same way that we have in the previous sections. Equation 3.14 connects the pinhole camera projection model to the dynamic motion of an object in the rotating coordinate frame of the sensor. So, inserting the equations above into the relations in Equation 3.14 will provide a differential equation that describes the motion of the object as it moves across the image plane. The $\dot{x}$ term in Equation 3.14 becomes

$$\dot{x} = -f\frac{-\rho_3\omega_2 + \rho_2\omega_3 + (\dot{\rho}_{nr})_1}{\rho_3} - x\frac{-\rho_2\omega_1 + \rho_1\omega_2 + (\dot{\rho}_{nr})_3}{\rho_3} \tag{3.78}$$

Expansion of the equation above yields

$$\dot{x} = f\omega_2 - f\frac{\rho_2\omega_3}{\rho_3} + x\frac{\rho_2\omega_1}{\rho_3} - x\frac{\rho_1\omega_2}{\rho_3} - f\frac{(\dot{\rho}_{nr})_1}{\rho_3} - x\frac{(\dot{\rho}_{nr})_3}{\rho_3} \tag{3.79}$$

Once again we will consider the relationship between the image plane coordinates and the relative position vector from Equation 2.4 to find

$$\dot{x} = f\left(\omega_2 - \frac{(\dot{\rho}_{nr})_1}{\rho_3}\right) - x\frac{(\dot{\rho}_{nr})_3}{\rho_3} + \omega_3 y + \frac{1}{f}\omega_2 x^2 - \frac{1}{f}\omega_1 xy \tag{3.80}$$

This relation is similar in structure to the rotation-only solution in Equation 3.22. In this case, however, we have not been able to fully remove references to the position vector, $\boldsymbol{\rho}$. This follows logically from the fact that we built this solution on the assumption that the object's velocity was available. Now, consider the equation for $\dot{y}$. Inserting the relations from Equation 3.77, we find

$$\dot{y} = -f\frac{\rho_3\omega_1 - \rho_1\omega_3 + (\dot{\rho}_{nr})_2}{\rho_3} - y\frac{-\rho_2\omega_1 + \rho_1\omega_2 + (\dot{\rho}_{nr})_3}{\rho_3} \tag{3.81}$$

The equation above expands to

$$\dot{y} = -f\omega_1 + f\frac{\rho_1\omega_3}{\rho_3} + y\frac{\rho_2\omega_1}{\rho_3} - y\frac{\rho_1\omega_2}{\rho_3} - f\frac{(\dot{\rho}_{nr})_2}{\rho_3} - y\frac{(\dot{\rho}_{nr})_3}{\rho_3} \tag{3.82}$$

and finally we arrive at

$$\dot{y} = -f\left(\omega_1 + \frac{(\dot{\rho}_{nr})_2}{\rho_3}\right) - \omega_3 x - y\frac{(\dot{\rho}_{nr})_3}{\rho_3} + \frac{1}{f}\omega_2 xy - \frac{1}{f}\omega_2 y^2 \tag{3.83}$$

through consideration of Equation 2.4. This Equation also contains explicit references to $\boldsymbol{\rho}$. Solving Equations 3.80 and 3.83 is non-trivial and will most likely require a numerical approach. We can proceed, however, by making the small field of view assumption that we used previously. Here we will also need the additional assumption that the range from the observer to the object changes slowly (i.e. ρ_3 is approximately constant). For distant objects this constraint is easily satisfied, however objects in low-Earth orbit may challenge this assumption. Define the terms

$$v_1 = \frac{(\dot{\rho}_{nr})_1}{\rho_3}, \quad v_2 = \frac{(\dot{\rho}_{nr})_2}{\rho_3}, \quad v_3 = \frac{(\dot{\rho}_{nr})_3}{\rho_3} \tag{3.84}$$

By our previous assumptions, these variables are approximately constant. Now, by substituting the terms above into Equations 3.80 and 3.83 and neglecting the higher-order terms, we have the system of Equations shown below.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \approx \begin{bmatrix} -v_3 & \omega_3 \\ -\omega_3 & -v_3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + f \begin{bmatrix} \omega_2 - v_1 \\ -\omega_1 - v_2 \end{bmatrix} \tag{3.85}$$

As in the rotation-only case, this system is linear in x and y and therefore is simple to solve. Further, we can take advantage of the structure of the first matrix to make this system easier to solve. The exponential of a sum of matrices satisfies the identity

$$\exp(A + B) = \exp(A) + \exp(B) \tag{3.86}$$

if the matrix multiplication of A and B is commutative (i.e. $AB = BA$) [59]. So, the matrix exponential of the first matrix multiplied by t is

$$\exp\left(\begin{bmatrix} -v_3 & \omega_3 \\ -\omega_3 & -v_3 \end{bmatrix} t\right) = \exp\left(\begin{bmatrix} 0 & \omega_3 \\ -\omega_3 & 0 \end{bmatrix} t + \begin{bmatrix} -v_3 & 0 \\ 0 & -v_3 \end{bmatrix} t\right) \tag{3.87}$$

The sum on the right-hand side is commutative, so the identity in Equation 3.86 holds and

$$\exp\left(\begin{bmatrix} -v_3 & \omega_3 \\ -\omega_3 & -v_3 \end{bmatrix} t\right) = \exp\left(\begin{bmatrix} 0 & \omega_3 \\ -\omega_3 & 0 \end{bmatrix} t\right) \exp\left(\begin{bmatrix} -v_3 & 0 \\ 0 & -v_3 \end{bmatrix} t\right) \tag{3.88}$$

We already know the first exponential term from Equation 3.56. The second term is simply

$$\exp\left(\begin{bmatrix} -v_3 & 0 \\ 0 & -v_3 \end{bmatrix} t\right) = \begin{bmatrix} e^{-v_3 t} & 0 \\ 0 & e^{-v_3 t} \end{bmatrix} = e^{-v_3 t} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.89}$$

So, the homogenous solution for Equation 3.85 is

$$\begin{bmatrix} x_h(t) \\ y_h(t) \end{bmatrix} = e^{-v_3 t} \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} \tag{3.90}$$

Now, we may find the full solution through the Variation of Constants formula.

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_h(t) \\ y_h(t) \end{bmatrix} + \int_0^t \exp\left(\begin{bmatrix} -v_3 & \omega_3 \\ -\omega_3 & -v_3 \end{bmatrix} (t - s)\right) f \begin{bmatrix} \omega_2 - v_1 \\ -\omega_1 - v_2 \end{bmatrix} ds \tag{3.91}$$

The latter two terms of the integral are constant, so we pull them outside. The solution of the integral term is

$$\int_0^t \exp\left(\begin{bmatrix} -v_3 & \omega_3 \\ -\omega_3 & -v_3 \end{bmatrix}(t-s)\right) ds =$$
$$\frac{fe^{-v_3 t}}{v_3^2+\omega_3^2}\begin{bmatrix} e^{v_3 t}v_3 - v_3\cos(\omega_3 t) + \omega_3\sin(\omega_3 t) & e^{v_3 t}\omega_3 - \omega_3\cos(\omega_3 t) - v_3\sin(\omega_3 t) \\ -e^{v_3 t}\omega_3 + \omega_3\cos(\omega_3 t) + v_3\sin(\omega_3 t) & e^{v_3 t}v_3 - v_3\cos(\omega_3 t) + \omega_3\sin(\omega_3 t) \end{bmatrix} \tag{3.92}$$

Thus, the solution for 3.85 is

$$\begin{aligned} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= e^{-v_3 t}\begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix}\begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} \\ &+ \frac{fv_3 e^{-v_3 t}}{v_3^2+\omega_3^2}\begin{bmatrix} e^{v_3 t} - \cos(\omega_3 t) & -\sin(\omega_3 t) \\ \sin(\omega_3 t) & e^{v_3 t} - \cos(\omega_3 t) \end{bmatrix}\begin{bmatrix} \omega_2 - v_1 \\ -\omega_1 - v_2 \end{bmatrix} \\ &+ \frac{f\omega_3 e^{-v_3 t}}{v_3^2+\omega_3^2}\begin{bmatrix} \sin(\omega_3 t) & e^{v_3 t} - \cos(\omega_3 t) \\ \cos(\omega_3 t) - e^{v_3 t} & \sin(\omega_3 t) \end{bmatrix}\begin{bmatrix} \omega_2 - v_1 \\ -\omega_1 - v_2 \end{bmatrix} \end{aligned} \tag{3.93}$$

This Equation is significantly more complicated than the solution for the rotation-only observer, but when $v_1 = v_2 = v_3 = 0$, the two solutions are equivalent. Further, in the case that v_3 is negligible, the equation above becomes

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix}\begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} + \frac{f}{\omega_3}\begin{bmatrix} \cos(\omega_3 t) - 1 & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) - 1 \end{bmatrix}\begin{bmatrix} \omega_1 + v_2 \\ \omega_2 - v_1 \end{bmatrix} \tag{3.94}$$

If the observer is not rotating, Equation 3.93 is equivalent to

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = e^{-v_3 t}\begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} - \frac{fe^{-v_3 t}}{v_3}\begin{bmatrix} e^{v_3 t} - 1 & 0 \\ 0 & e^{v_3 t} - 1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{3.95}$$

Though the Equation above is undefined for $v_3 = 0$, the limit of the second term as v_3 approaches zero is

$$\lim_{v_3 \to 0}\left(\frac{e^{v_3 t} - 1}{v_3}\right) = t \tag{3.96}$$

So, with $v_3 = 0$, we have

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} - f\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} t \tag{3.97}$$

The result above describes the motion of an object across the image plane of a non-rotating sensor with an image-plane velocity vector of $(v_1, v_2)^T$.

3.1.5 Non-Rotating Observer, Constant Acceleration Object

Another possible observation scenario involves a non-rotating sensor with an object passing through the view. This situation might occur when the sensor is tasked on some prior knowledge in a "step, stare" mode, where the sensor points at the location on the celestial sphere where the target object will be at some point in the future. In the frame of a non-rotating sensor, the differential equation describing the motion of an object undergoing constant acceleration is

$$\ddot{\boldsymbol{\rho}} = \boldsymbol{a} \tag{3.98}$$

From the individual components of $\ddot{\rho}$, we see that the following relations hold true.

$$\begin{aligned} \ddot{\rho}_1 &= a_1 & &\Longrightarrow & \dot{\rho}_1 &= a_1 t + \dot{\rho}_1 \left(t_0\right) \\ \ddot{\rho}_2 &= a_2 & &\Longrightarrow & \dot{\rho}_2 &= a_2 t + \dot{\rho}_2 \left(t_0\right) \\ \ddot{\rho}_3 &= a_3 & &\Longrightarrow & \dot{\rho}_3 &= a_3 t + \dot{\rho}_3 \left(t_0\right) \end{aligned} \tag{3.99}$$

At this point we will deviate slightly from the previous derivations and, instead of using the first derivatives of the pinhole model in Equation 3.14, we will use the second derivatives from Equation 3.15. Though the equations above provide the first derivatives of $\dot{\rho}$ and allow the use of Equation 3.14, the second derivative formulation will provide a more intuitive structure. Recall that the second derivatives of the pinhole projection model are

$$\begin{aligned} \ddot{x} &= -2\dot{x}\frac{\dot{\rho}_3}{\rho_3} - x\frac{\ddot{\rho}_3}{\rho_3} - f\frac{\ddot{\rho}_1}{\rho_3} \\ \ddot{y} &= -2\dot{y}\frac{\dot{\rho}_3}{\rho_3} - y\frac{\ddot{\rho}_3}{\rho_3} - f\frac{\ddot{\rho}_2}{\rho_3} \end{aligned} \tag{3.100}$$

Beginning with the $\ddot{x}$ equation, we may insert the relations for $\ddot{\rho}_1$, $\ddot{\rho}_2$, and $\dot{\rho}_3$ to find

$$\ddot{x} = -2\dot{x}\frac{a_3 t + \dot{\rho}_3 \left(t_0\right)}{\rho_3} - x\frac{a_3}{\rho_3} - f\frac{a_1}{\rho_3} \tag{3.101}$$

Similarly, the $\ddot{y}$ equation becomes

$$\ddot{y} = -2\dot{y}\frac{a_3 t + \dot{\rho}_3 \left(t_0\right)}{\rho_3} - y\frac{a_3}{\rho_3} - f\frac{a_2}{\rho_3} \tag{3.102}$$

Now, consider the previous two equations in matrix form.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2\frac{a_3 t + \dot{\rho}_3(t_0)}{\rho_3} & 0 \\ 0 & 0 & 0 & -2\frac{a_3 t + \dot{\rho}_3(t_0)}{\rho_3} \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} - \frac{f}{\rho_3}\begin{bmatrix} 0 \\ 0 \\ a_1 \\ a_2 \end{bmatrix} \tag{3.103}$$

The terms in the lower right-hand corner of the matrix coefficient includes the ratio between the range rate and the range. We encountered this relationship in the previous section as well. If we make the additional assumption that the range rate is small compared to the overall range, then the equation above becomes

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} - \frac{f}{\rho_3} \begin{bmatrix} 0 \\ 0 \\ a_1 \\ a_2 \end{bmatrix} \tag{3.104}$$

If we take this assumption one step further and assume that ρ_3 changes slowly, then the latter term in the equation above is approximately constant. So, planar translational motion provides an approximation of the motion of the image-plane projection of an object moving with constant acceleration. Indeed, motion models which assume smooth motion with small accelerations are commonly employed in point tracking literature [60, 61]. For very distant objects ($\rho_3 \gg a_1$ and $\rho_3 \gg a_2$), the system above approximately describes translational motion with constant velocity.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{3.105}$$

The solution of the differential equation above is

$$\begin{bmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(t_0) \\ y(t_0) \\ \dot{x}(t_0) \\ \dot{y}(t_0) \end{bmatrix} \tag{3.106}$$

3.2 Motion in the Distortion-Corrected Image Plane

Though the pinhole model is a convenient theoretical starting point for for analysis of optical sensors, it is only an approximate model and often falls short of capturing the behavior of real-world sensors. It is for this reason that distortion correction techniques, like those discussed in Chapter One, have been standardized. The SIP technique provides a standard way to transform image-plane coordinates into some other coordinates system of interest. Which coordinate system results is dependent on the way that the user formulates the correction polynomial. The Astrometry.net software package, for example, transforms the pixel coordinates to the right ascension and declination frame [18]. Here we will consider a more simple model which provides clean analytic results and relates more easily to the derivations in the previous section.

Assume that the a distortion polynomial is available that transforms pixel locations into a coordinate system wherein each pixel is separated by equal angular steps along the x and y

axes. Choose

$$\begin{aligned} x &= c\theta \\ y &= c\phi \end{aligned} \tag{3.107}$$

where c is a scaling parameter and θ and ϕ describe the orientation of the object's position vector relative to the boresight of the sensor. These are the same θ and ϕ values defined in the previous section. Figure 3.3 in the previous section illustrates the geometry for ϕ, which is the same as for θ only with Δy instead of Δx.

Let us determine a choice for the scaling parameter, c. Though we can use an arbitrary scaling parameter, it will be beneficial to choose one which relates more closely with the previous derivations. Recall that the pinhole model is

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\frac{f}{\rho_3}\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \tag{3.108}$$

Also recall that the angles θ and ϕ are defined as

$$\begin{aligned} \tan\theta &= \frac{\rho_1}{\rho_3} = -\frac{x}{f} \\ \tan\phi &= \frac{\rho_2}{\rho_3} = -\frac{y}{f} \end{aligned} \tag{3.109}$$

These angles describe the orientation of the position vector in the $x - z$ and $y - z$ planes, respectively. Rearranging the equations above, we find

$$\begin{aligned} x &= -f\tan\theta \\ y &= -f\tan\phi \end{aligned} \tag{3.110}$$

Under a small angle assumption, the following are true.

$$\tan\theta \approx \theta, \qquad \tan\phi \approx \phi \tag{3.111}$$

So, as θ and ϕ approach zero,

$$\begin{aligned} x &\approx -f\theta \\ y &\approx -f\phi \end{aligned} \tag{3.112}$$

Thus, if we choose $c = -f$, then the model

$$\begin{aligned} x &= -f\theta \\ y &= -f\phi \end{aligned} \tag{3.113}$$

aligns better with the pinhole camera model for objects closer to alignment with the boresight of the sensor. For objects exactly along the boresight, the two models agree perfectly. Further, this result implies that small field of view sensors that adhere to the pinhole model will also approximately adhere to this formulation of the distortion-corrected model. So we will proceed with the distortion corrected model above. Any image analyzed with this approach must be corrected with a distortion polynomial that results in the model above. Though the distortion correction will likely not be perfect, we assume that the resulting errors are small.

Before moving on it is worth considering how one would correct an image from a sensor adhering to the pinhole model so that it instead satisfies Equation 3.113. A correction polynomial is easily available from consideration of Equation 3.110. Define the distortion-corrected coordinates as x_d and y_d. Distortion corrections are usually performed by modifying the focal plane coordinate system with some distortion polynomial [37, 38]. Define the distortion-corrected x coordinateas

$$x = x_d + \eta\left(x_d\right) \tag{3.114}$$

where $\eta\left(x_d\right)$ is the distortion polynomial that converts from coordinates in a sensor adhering to the pinhole model to those of the distortion-corrected model in Equation 3.113. Consider the choice of $\eta\left(x_d\right)$ below.

$$\eta\left(x_d\right) = -x_d - f\tan^{-1}\left(-\frac{x_d}{f}\right) \tag{3.115}$$

This correction effectively cancels out the existing distorted x_d coordinate and replaces it with the focal length multiplied by the angle ϕ. Inserting $\eta\left(x_d\right)$ into the first equation, we have

$$x = x_d - x_d - f\tan^{-1}\left(-\frac{x_d}{f}\right) \tag{3.116}$$

From Equation 3.110, $x_d = -f\tan\theta$, so the equation above reduces to

$$x = -f\tan^{-1}\left(\frac{f\tan\theta}{f}\right) = -f\theta \tag{3.117}$$

Thus, the choice of distortion correction in Equation 3.115 corrects an image from a pinhole sensor to satisfy Equation 3.113. The same can be shown for the y coordinate of the distortion-corrected model. Define

$$y = y_d + \xi\left(y_d\right) \tag{3.118}$$

where $\xi\left(y_d\right)$ is the distortion correction for the y_d coordinate. Now consider a choice for $\xi\left(y_d\right)$ similar to that of $\eta\left(y_d\right)$.

$$\xi\left(y_d\right) = -y_d - f\tan^{-1}\left(-\frac{y_d}{f}\right) \tag{3.119}$$

Again, this correction cancels out the existing coordinate and replaces it with $-f\phi$.

$$y = y_d - y_d - f\tan^{-1}\left(-\frac{y_d}{f}\right) = -f\tan^{-1}\left(\frac{f\tan\phi}{f}\right) = -f\phi \tag{3.120}$$

It is clear that neither the choice of $\eta\left(x_d\right)$ nor the choice of $\xi\left(y_d\right)$ are polynomials. Both correction functions can, however, be approximated as polynomials. The Taylor series of $\tan^{-1} g$ for some arbitrary g is [62]

$$\tan^{-1} g = \sum_{n=0}^{\infty} \frac{\left(-1\right)^n}{2n+1} g^{2n+1} \tag{3.121}$$

So, applying the Taylor series expansion, $\eta\left(x_d\right)$ is

$$\eta\left(x_d\right) = -x_d - f \sum_{n=0}^{\infty} \frac{\left(-1\right)^n}{2n+1} \left(-\frac{x_d}{f}\right)^{2n+1} \tag{3.122}$$

or, since the $-x_d$ term cancels with the first term of the series,

$$\eta\left(x_d\right) = -f \sum_{n=1}^{\infty} \frac{\left(-1\right)^n}{2n+1} \left(-\frac{x_d}{f}\right)^{2n+1} \tag{3.123}$$

The first five terms of $\eta\left(x_d\right)$ are

$$\eta\left(x_d\right) = -\frac{1}{3f^2}x_d^3 + \frac{1}{5f^4}x_d^5 - \frac{1}{7f^6}x_d^7 + \frac{1}{9f^8}x_d^9 - \frac{1}{11f^{10}}x_d^{11} + \cdots \tag{3.124}$$

and the first five terms of $\xi\left(y_d\right)$ are

$$\xi\left(y_d\right) = -\frac{1}{3f^2}y_d^3 + \frac{1}{5f^4}y_d^5 - \frac{1}{7f^6}y_d^7 + \frac{1}{9f^8}y_d^9 - \frac{1}{11f^{10}}y_d^{11} + \cdots \tag{3.125}$$

The relations above give us a way to correct an image created by a sensor adhering to the pinhole camera model to fit the desired model in Equation 3.113. Since we have formulated the correction as a polynomial, it meshes well with the industry-standard way of describing the distortion in a sensor discussed in Chapter 2, Section 2.1. In general, a sensor will have additional distortions not modeled here. In those cases, it is possible to estimate the distortion polynomial through a star identification process [18]. Regardless of how the distortion polynomial is computed, we are able to correct an image and now we will the derive the new motion models that result. Considering the relations between θ, ϕ, and $\boldsymbol{\rho}$ in Equation 3.110, we can write the distortion-corrected projection equations as

$$\begin{aligned} x &= -f \tan^{-1}\left(\frac{\rho_1}{\rho_3}\right) \\ y &= -f \tan^{-1}\left(\frac{\rho_2}{\rho_3}\right) \end{aligned} \tag{3.126}$$

We will follow a similar procedure to the one in the previous section. Since we will not necessarily have an explicit solution for $\boldsymbol{\rho}\left(t\right)$, it is necessary to find a connection between the projection equation and higher-order equations of motion. Taking the derivative of x and y results in

$$\begin{aligned} \dot{x} &= -f\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\upsilon}_3}{\rho_3^2}\right)\left(1+\frac{\rho_1^2}{\rho_3^2}\right)^{-1} \\ \dot{y} &= -f\left(\frac{\dot{\rho}_2}{\rho_3} - \frac{\rho_2\dot{\upsilon}_3}{\rho_3^2}\right)\left(1+\frac{\rho_2^2}{\rho_3^2}\right)^{-1} \end{aligned} \tag{3.127}$$

Since the derivative of the inverse tangent function is

$$\frac{d}{dt}\tan^{-1}\left(g\left(t\right)\right) = \frac{\dot{g}\left(t\right)}{1+\left[g\left(t\right)\right]^2} \tag{3.128}$$

where $g\left(t\right)$ is an arbitrary function of t. We know that $\frac{\rho_1}{\rho_3} = \tan\theta$ so

$$\left(1+\frac{\rho_1^2}{\rho_3^2}\right)^{-1} = \left(1+\tan^2\theta\right)^{-1} \tag{3.129}$$

Likewise, since $\frac{\rho_2}{\rho_3} = \tan\phi$

$$\left(1+\frac{\rho_2^2}{\rho_3^2}\right)^{-1} = \left(1+\tan^2\phi\right)^{-1} \tag{3.130}$$

So the first derivatives of the distortion corrected model are

$$\dot{x} = -\frac{f}{1+\tan^2\theta}\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\rho}_3}{\rho_3^2}\right) \tag{3.131}$$

and

$$\dot{y} = -\frac{f}{1+\tan^2\phi}\left(\frac{\dot{\rho}_2}{\rho_3} - \frac{\rho_2\dot{\rho}_3}{\rho_3^2}\right) \tag{3.132}$$

Through trigonometric identities, we may also write the equations above as

$$\begin{aligned} \dot{x} &= -f\cos^2\theta\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\rho}_3}{\rho_3^2}\right) \\ \dot{y} &= -f\cos^2\phi\left(\frac{\dot{\rho}_2}{\rho_3} - \frac{\rho_2\dot{\rho}_3}{\rho_3^2}\right) \end{aligned} \tag{3.133}$$

Further, inserting Equation 3.113 produces

$$\begin{aligned} \dot{x} &= -f\cos^2\left(\frac{x}{f}\right)\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\rho}_3}{\rho_3^2}\right) \\ \dot{y} &= -f\cos^2\left(\frac{y}{f}\right)\left(\frac{\dot{\rho}_2}{\rho_3} - \frac{\rho_2\dot{\rho}_3}{\rho_3^2}\right) \end{aligned} \tag{3.134}$$

As was the case in the analysis for the pinhole model, an explicit first derivative may not be available for $\boldsymbol{\rho}$. In these cases the second derivative of the projection model may provide a better connection between the projection equation and the relative dynamics. The derivative of the $\dot{x}$ equation with respect to time is

$$\ddot{x} = f\frac{2\cos\left(\frac{x}{f}\right)\sin\left(\frac{x}{f}\right)\dot{x}}{f}\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\rho}_3}{\rho_3^2}\right) + f\cos^2\left(\frac{x}{f}\right)\left(2\frac{\dot{\rho}_1\dot{\rho}_3}{\rho_3^2} - 2\frac{\rho_1\dot{\rho}_3^2}{\rho_3^3} - \frac{\ddot{\rho}_1}{\rho_3} + \frac{\rho_1\ddot{\rho}_3}{\rho_3^2}\right) \tag{3.135}$$

Expanded, the equation above is

$$\begin{aligned}\ddot{x} =& 2\dot{x}\cos\left(\frac{x}{f}\right)\sin\left(\frac{x}{f}\right)\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\rho}_3}{\rho_3^2}\right) + f\cos^2\left(\frac{x}{f}\right)\left(2\frac{\dot{\rho}_1\dot{\rho}_3}{\rho_3^2} - 2\frac{\rho_1\dot{\rho}_3^2}{\rho_3^3}\right) \\ &+ f\cos^2\left(\frac{x}{f}\right)\left(-\frac{\ddot{\rho}_1}{\rho_3} + \frac{\rho_1\ddot{\rho}_3}{\rho_3^2}\right)\end{aligned} \tag{3.136}$$

Further, we can insert the relations that we know for $\dot{x}$ to find

$$\ddot{x} = -\frac{2}{f}\tan\left(\frac{x}{f}\right)\dot{x}^2 - 2f\frac{\dot{\rho}_3}{\rho_3}\dot{x} - f\cos^2\left(\frac{x}{f}\right)\left(\frac{\ddot{\rho}_1}{\rho_3} - \frac{\rho_1\ddot{\rho}_3}{\rho_3^2}\right) \tag{3.137}$$

An identical derivation follows for the y term. The derivative of the $\dot{y}$ equation with respect to time is

$$\ddot{y} = f\frac{2\cos\left(\frac{y}{f}\right)\sin\left(\frac{y}{f}\right)\dot{y}}{f}\left(\frac{\dot{\rho}_2}{\rho_3} - \frac{\rho_2\dot{\rho}_3}{\rho_3^2}\right) + f\cos^2\left(\frac{y}{f}\right)\left(2\frac{\dot{\rho}_2\dot{\rho}_3}{\rho_3^2} - 2\frac{\rho_2\dot{\rho}_3^2}{\rho_3^3} - \frac{\ddot{\rho}_2}{\rho_3} + \frac{\rho_2\ddot{\rho}_3}{\rho_3^2}\right) \tag{3.138}$$

Again, we expand the equation above to find

$$\begin{aligned}\ddot{y} =& 2\dot{y}\cos\left(\frac{y}{f}\right)\sin\left(\frac{y}{f}\right)\left(\frac{\dot{\rho}_2}{\rho_3} - \frac{\rho_2\dot{\rho}_3}{\rho_3^2}\right) + f\cos^2\left(\frac{y}{f}\right)\left(2\frac{\dot{\rho}_2\dot{\rho}_3}{\rho_3^2} - 2\frac{\rho_2\dot{\rho}_3^2}{\rho_3^3}\right) \\ &+ f\cos^2\left(\frac{y}{f}\right)\left(-\frac{\ddot{\rho}_2}{\rho_3} + \frac{\rho_2\ddot{\rho}_3}{\rho_3^2}\right)\end{aligned} \tag{3.139}$$

Finally, inserting the equation for $\dot{y}$ produces

$$\ddot{y} = -\frac{2}{f}\tan\left(\frac{y}{f}\right)\dot{y}^2 - 2f\frac{\dot{\rho}_3}{\rho_3}\dot{y} - f\cos^2\left(\frac{y}{f}\right)\left(\frac{\ddot{\rho}_2}{\rho_3} - \frac{\rho_2\ddot{\rho}_3}{\rho_3^2}\right) \tag{3.140}$$

So, in summary, the first derivatives of the distortion-correct model are

$$\begin{aligned}\dot{x} &= -f\cos^2\left(\frac{x}{f}\right)\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\rho}_3}{\rho_3^2}\right) \\ \dot{y} &= -f\cos^2\left(\frac{y}{f}\right)\left(\frac{\dot{\rho}_2}{\rho_3} - \frac{\rho_2\dot{\rho}_3}{\rho_3^2}\right)\end{aligned} \tag{3.141}$$

and the second derivatives are

$$\begin{aligned}\ddot{x} &= -\frac{2}{f}\tan\left(\frac{x}{f}\right)\dot{x}^2 - 2f\frac{\dot{\rho}_3}{\rho_3}\dot{x} - f\cos^2\left(\frac{x}{f}\right)\left(\frac{\ddot{\rho}_1}{\rho_3} - \frac{\rho_1\ddot{\rho}_3}{\rho_3^2}\right)\\ \ddot{y} &= -\frac{2}{f}\tan\left(\frac{y}{f}\right)\dot{y}^2 - 2f\frac{\dot{\rho}_3}{\rho_3}\dot{y} - f\cos^2\left(\frac{y}{f}\right)\left(\frac{\ddot{\rho}_2}{\rho_3} - \frac{\rho_2\ddot{\rho}_3}{\rho_3^2}\right)\end{aligned} \tag{3.142}$$

These differential equations relate the image plane coordinates in a distortion-corrected image to the relative dynamics of an observed object. When the distortion in an optical system causes it to deviate significantly from the pinhole model it may be beneficial to correct the image with a distortion polynomial and use these equations instead.

3.2.1 Constant Rotation Observer, Inertially-Fixed Object

Consider again the case of an observer rotating with a constant angular rate. Recall that the components of the relative position vector $\boldsymbol{\rho}$ evolve according to

$$\begin{aligned}\dot{\rho}_1 &= \omega_3\rho_2 - \omega_2\rho_3\\ \dot{\rho}_2 &= -\omega_3\rho_1 + \omega_1\rho_3\\ \dot{\rho}_3 &= \omega_2\rho_1 - \omega_1\rho_2\end{aligned} \tag{3.143}$$

The first derivative equations for x and y provide a connection between the distortion-corrected projection model and the relative position dynamics above. This analysis will follow an identical procedure to the one in the previous section for the pinhole model. In fact, since the distortion-corrected projection model in Equation 3.113 was designed to relate closely to the pinhole model at small angles, the resulting equations will have very similar structure. Beginning with the equation for $\dot{x}$, the equations above show that

$$\dot{x} = -f\cos^2\left(\frac{x}{f}\right)\left(\frac{\omega_3\rho_2 - \omega_2\rho_3}{\rho_3} - \frac{\rho_1\left(\omega_2\rho_1 - \omega_1\rho_2\right)}{\rho_3^2}\right) \tag{3.144}$$

which expands to

$$\dot{x} = -f\cos^2\left(\frac{x}{f}\right)\left(\frac{\omega_3\rho_2}{\rho_3} - \frac{\omega_2\rho_3}{\rho_3} - \frac{\omega_2\rho_1^2}{\rho_3^2} + \frac{\omega_1\rho_1\rho_2}{\rho_3^2}\right) \tag{3.145}$$

Recalling the relations between x, y, and $\boldsymbol{\rho}$ from Equation 3.126,

$$\dot{x} = -f\cos^2\left(\frac{x}{f}\right)\left(-\omega_3\tan\left(\frac{y}{f}\right) - \omega_2 - \omega_2\tan^2\left(\frac{x}{f}\right) + \omega_1\tan\left(\frac{x}{f}\right)\tan\left(\frac{y}{f}\right)\right) \tag{3.146}$$

This equation has a very similar structure to that in Equation 3.22 with tangent terms in place of the regular variables. For $\dot{y}$ we have

$$\dot{y} = -f\cos^2\left(\frac{y}{f}\right)\left(\frac{-\omega_3\rho_1 + \omega_1\rho_3}{\rho_3} - \frac{\rho_2\left(\omega_2\rho_1 - \omega_1\rho_2\right)}{\rho_3^2}\right) \tag{3.147}$$

which rearranges to

$$\dot{y} = -f\cos^2\left(\frac{y}{f}\right)\left(-\frac{\omega_3\rho_1}{\rho_3} + \frac{\omega_1\rho_3}{\rho_3} - \frac{\omega_2\rho_1\rho_2}{\rho_3^2} + \frac{\omega_1\rho_2^2}{\rho_3^2}\right) \tag{3.148}$$

and finally becomes

$$\dot{y} = -f\cos^2\left(\frac{y}{f}\right)\left(\omega_3\tan\left(\frac{x}{f}\right) + \omega_1 - \omega_2\tan\left(\frac{x}{f}\right)\tan\left(\frac{y}{f}\right) + \omega_1\tan^2\left(\frac{y}{f}\right)\right) \tag{3.149}$$

Notice that if we make a small-angle assumption such that $\cos\left(\frac{x}{f}\right) = \cos\left(\frac{y}{f}\right) \approx 1$, $\tan\left(-\frac{x}{f}\right) \approx -\frac{x}{f}$, and $\tan\left(-\frac{y}{f}\right) \approx -\frac{y}{f}$, then we see that the equations for $\dot{x}$ and $\dot{y}$ become

$$\dot{x} = -f\left[-\omega_3\frac{y}{f} - \omega_2 - \omega_2\frac{x^2}{f^2} + \omega_1\frac{xy}{f^2}\right] = f\omega_3 y + f\omega_2 + \omega_2\frac{x^2}{f} - \omega_1\frac{xy}{f} \tag{3.150}$$

and

$$\dot{y} = -f\left[\omega_3\frac{x}{f} + \omega_1 - \omega_2\frac{xy}{f^2} + \omega_1\frac{y^2}{f^2}\right] = -\omega_3 x - f\omega_1 + \omega_2\frac{xy}{f} - \omega_1\frac{y^2}{f} \tag{3.151}$$

which are identical to the results for the pinhole model. Thus, if $x \ll f$ and $y \ll f$, then the two models are approximately equal. This result agrees with the initial choice to define our distortion-corrected model as one which aligns with the pinhole model for small angles. If the small angle assumption holds, then all of the same solutions from the previous section apply for the distortion-corrected model as well. See Equations 3.46, 3.49, and 3.60 for these solutions.

3.2.2 Constant Rotation Observer, Constant Velocity Object

This section derives the equations of motion for an object with constant velocity moving across the image plane of an observer with a constant rotation rate according to the distortion-corrected projection model. Recall that the relative position vector, $\boldsymbol{\rho}$ evolves according to [58]

$$\dot{\boldsymbol{\rho}} = \dot{\boldsymbol{\rho}}_{nr} - \boldsymbol{\omega} \times \boldsymbol{\rho} \tag{3.152}$$

where $\dot{\boldsymbol{\rho}}_{nr}$ is the velocity in the inertial frame of a particle relative to the sensor. The individual components of $\dot{\boldsymbol{\rho}}_{nr}$ are

$$\begin{aligned}\dot{\rho}_1 &= (\dot{\rho}_{nr})_1 + \omega_3\rho_2 - \omega_2\rho_3\\ \dot{\rho}_2 &= (\dot{\rho}_{nr})_2 - \omega_3\rho_1 + \omega_1\rho_3\\ \dot{\rho}_3 &= (\dot{\rho}_{nr})_3 + \omega_2\rho_1 - \omega_1\rho_2\end{aligned} \tag{3.153}$$

From Equation 3.141, the relationship between $\dot{x}$ and $\boldsymbol{\rho}$ is

$$\dot{x} = -f\cos^2\left(\frac{x}{f}\right)\left(\frac{\dot{\rho}_1}{\rho_3} - \frac{\rho_1\dot{\rho}_3}{\rho_3^2}\right) \tag{3.154}$$

Combining the equation above with Equation 3.153, we find

$$\dot{x} = -f\cos^2\left(\frac{x}{f}\right)\left(\frac{(\dot{\rho}_{nr})_1 + \omega_3\rho_2 - \omega_2\rho_3}{\rho_3} - \rho_1\frac{(\dot{\rho}_{nr})_3 + \omega_2\rho_1 - \omega_1\rho_2}{\rho_3^2}\right) \tag{3.155}$$

which expands to

$$\dot{x} = -f\cos^2\left(\frac{x}{f}\right)\left(-\omega_2 + \frac{(\dot{\rho}_{nr})_1}{\rho_3} - \frac{\rho_1}{\rho_3^2}(\dot{\rho}_{nr})_3 + \frac{\rho_2}{\rho_3}\omega_3 - \frac{\rho_1^2}{\rho_3^2}\omega_2 + \frac{\rho_1\rho_2}{\rho_3^2}\omega_1\right) \tag{3.156}$$

Finally, recall the relationship between $\boldsymbol{\rho}$ and the image plane variables x and y from Equation 3.126. Combining these relations with the equation above produces

$$\begin{aligned}\dot{x} = -f\cos^2\left(\frac{x}{f}\right)\Bigg(-\omega_2 + \frac{(\dot{\rho}_{nr})_1}{\rho_3} + \tan\left(\frac{x}{f}\right)\frac{(\dot{\rho}_{nr})_3}{\rho_3} - \tan\left(\frac{y}{f}\right)\omega_3 \\ - \tan^2\left(\frac{x}{f}\right)\omega_2 + \tan\left(\frac{x}{f}\right)\tan\left(\frac{y}{f}\right)\omega_1\Bigg)\end{aligned} \tag{3.157}$$

As in the previous section, under the small angle assumption ($x \ll f$ and $y \ll f$) the equation above is approximately equal to the pinhole model solution from Equation 3.80. Therefore the solutions from that section also apply if the small angle assumptions are satisfied. An identical derivation exists for the $\dot{y}$ term of Equation 3.141. Recall that $\dot{y}$ is

$$\dot{y} = -f\cos^2\left(\frac{y}{f}\right)\left(\frac{(\dot{\rho}_{nr})_2 - \omega_3\rho_1 + \omega_1\rho_3}{\rho_3} - \rho_2\frac{(\dot{\rho}_{nr})_3 + \omega_2\rho_1 - \omega_1\rho_2}{\rho_3^2}\right) \tag{3.158}$$

Inserting Equation 3.153 leads to

$$\dot{y} = -f\cos^2\left(\frac{x}{f}\right)\left(\omega_1 + \frac{(\dot{\rho}_{nr})_2}{\rho_3} - \frac{\rho_1}{\rho_3}\omega_3 - \frac{\rho_2}{\rho_3^2}(\dot{\rho}_{nr})_3 - \frac{\rho_1\rho_2}{\rho_3^2}\omega_2 + \frac{\rho_2^2}{\rho_3^2}\omega_1\right) \tag{3.159}$$

and finally, from the relationship between $\boldsymbol{\rho}$ and the image plane coordinates, we have

$$\begin{aligned}\dot{y} = -f\cos^2\left(\frac{x}{f}\right)\Bigg(\omega_1 + \frac{(\dot{\rho}_{nr})_2}{\rho_3} + \tan\left(\frac{x}{f}\right)\omega_3 + \tan\left(\frac{y}{f}\right)\frac{(\dot{\rho}_{nr})_3}{\rho_3} \\ - \tan\left(\frac{x}{f}\right)\tan\left(\frac{y}{f}\right)\omega_2 + \tan^2\left(\frac{y}{f}\right)\omega_1\Bigg)\end{aligned} \tag{3.160}$$

Under the small angle assumption, this equation is approximately equal to the pinhole model result from Equation 3.83.

3.2.3 Non-Rotating Observer, Constant Acceleration Object

Now consider the case of the non-rotating observer viewing an object moving with constant acceleration in the reference frame of the sensor. The differential equation describing the evolution of the relative position vector is

$$\ddot{\boldsymbol{\rho}} = \boldsymbol{a} \tag{3.161}$$

where $\boldsymbol{a}$ denotes the acceleration vector. Recall that the individual components of $\ddot{\boldsymbol{\rho}}$ show that

$$\begin{aligned} \ddot{\rho}_1 &= a_1 && \Longrightarrow && \dot{\rho}_1 = a_1 t + \dot{\rho}_1 (t_0) \\ \ddot{\rho}_2 &= a_2 && \Longrightarrow && \dot{\rho}_2 = a_2 t + \dot{\rho}_2 (t_0) \\ \ddot{\rho}_3 &= a_3 && \Longrightarrow && \dot{\rho}_3 = a_3 t + \dot{\rho}_3 (t_0) \end{aligned} \tag{3.162}$$

Recall from Equation 3.142 that the second derivative with respect to time of the x coordinate in the distortion-corrected model is

$$\ddot{x} = -\frac{2}{f} \tan\left(\frac{x}{f}\right) \dot{x}^2 - 2f \frac{\dot{\rho}_3}{\rho_3} \dot{x} - f \cos^2\left(\frac{x}{f}\right) \left(\frac{\ddot{\rho}_1}{\rho_3} - \frac{\rho_1 \ddot{\rho}_3}{\rho_3^2}\right) \tag{3.163}$$

Inserting the relations for $\ddot{\boldsymbol{\rho}}$ from Equation 3.162 yields

$$\ddot{x} = -\frac{2}{f} \tan\left(\frac{x}{f}\right) \dot{x}^2 - 2f \frac{a_3 t + \dot{\rho}_3 (t_0)}{\rho_3} \dot{x} - f \cos^2\left(\frac{x}{f}\right) \left(\frac{a_1}{\rho_3} - \frac{\rho_1 a_3}{\rho_3^2}\right) \tag{3.164}$$

From the relationship between $\boldsymbol{\rho}$ and the image plane coordinates in Equation 3.126, the equation above is equivalent to

$$\ddot{x} = -\frac{2}{f} \tan\left(\frac{x}{f}\right) \dot{x}^2 - 2f \frac{a_3 t + \dot{\rho}_3 (t_0)}{\rho_3} \dot{x} - f \cos^2\left(\frac{x}{f}\right) \left(\frac{a_1}{\rho_3} + \tan\left(\frac{x}{f}\right) \frac{a_3}{\rho_3}\right) \tag{3.165}$$

Similarly, recall that the second derivative with respect to time of the y coordinate of the distortion corrected model is

$$\ddot{y} = -\frac{2}{f} \tan\left(\frac{y}{f}\right) \dot{y}^2 - 2f \frac{\dot{\rho}_3}{\rho_3} \dot{y} - f \cos^2\left(\frac{y}{f}\right) \left(\frac{\ddot{\rho}_2}{\rho_3} - \frac{\rho_2 \ddot{\rho}_3}{\rho_3^2}\right) \tag{3.166}$$

After combining the equation above with Equation 3.162, it becomes

$$\ddot{y} = -\frac{2}{f} \tan\left(\frac{y}{f}\right) \dot{y}^2 - 2f \frac{a_3 t + \dot{\rho}_3 (t_0)}{\rho_3} \dot{y} - f \cos^2\left(\frac{y}{f}\right) \left(\frac{a_2}{\rho_3} - \frac{\rho_2 a_3}{\rho_3^2}\right) \tag{3.167}$$

Then, from the relationship between $\boldsymbol{\rho}$ and y, we have

$$\ddot{y} = -\frac{2}{f} \tan\left(\frac{y}{f}\right) \dot{y}^2 - 2f \frac{a_3 t + \dot{\rho}_3 (t_0)}{\rho_3} \dot{y} - f \cos^2\left(\frac{y}{f}\right) \left(\frac{a_2}{\rho_3} + \tan\left(\frac{y}{f}\right) \frac{a_3}{\rho_3}\right) \tag{3.168}$$

Due to the corrected projection model, these solutions are more complicated than those for the pinhole model. We can, however, still make some approximations. Consider again the assumption that the range, ρ_3, is large. The last three terms of the $\ddot{x}$ and $\ddot{y}$ contain the range in the denominator and therefore may be neglected in some cases. The second term has been neglected in previous sections, under the assumption that the rate of change of the range is much smaller than the range itself. The same applies to the third term: if the accelerations are much smaller than the range, then they may be neglected as well. So, if we neglect these two terms, we arrive at

$$\begin{aligned} \ddot{x} &= -\frac{2}{f}\tan\left(\frac{x}{f}\right)\dot{x}^2 \\ \ddot{y} &= -\frac{2}{f}\tan\left(\frac{y}{f}\right)\dot{y}^2 \end{aligned} \tag{3.169}$$

The remaining term is the key difference between this result and the results for the pinhole model. The velocity terms, $\dot{x}$ and $\dot{y}$ are often small for detectable objects (i.e. objects that stay in view for multiple frames), so the square of the velocity may be safely approximated as zero in some cases. In those situations, the motion model is simply constant-velocity translational motion. If we neglect the first term and keep the remaining terms of Equations 3.165 and 3.168, we have the differential equations

$$\begin{aligned} \ddot{x} &= -2f\frac{a_3 t + \dot{\rho}_3\left(t_0\right)}{\rho_3}\dot{x} - f\cos^2\left(\frac{x}{f}\right)\left(\frac{a_1}{\rho_3} + \tan\left(\frac{x}{f}\right)\frac{a_3}{\rho_3}\right) \\ \ddot{y} &= -2f\frac{a_3 t + \dot{\rho}_3\left(t_0\right)}{\rho_3}\dot{y} - f\cos^2\left(\frac{y}{f}\right)\left(\frac{a_2}{\rho_3} + \tan\left(\frac{y}{f}\right)\frac{a_3}{\rho_3}\right) \end{aligned} \tag{3.170}$$

which are structurally similar to the results for the pinhole model from Equations 3.101 and 3.102. Indeed, the small angle assumption that $x \ll f$ and $y \ll f$ leads to an equivalent result.

3.3 Geometric Image-Plane Star Path Description

Here we will derive explicit equations describing the path of an inertially-fixed object along the focal plane of a rotating camera. This approach will rely on the assumption that the sensor rotates with a constant axis of rotation. Note that this assumption is somewhat different from those in the previous sections in that it is acceptable for the magnitude of the angular rate to vary as long as the rotation axis remains fixed. The derivation here will rely on the pin-hole camera model, described by the Collinearity equation [35]. In this case, however, we will think about the projection geometrically rather than algebraically.

With a fixed axis of rotation, inertially-fixed objects trace circles in the rotating frame of the sensor around that axis. With this knowledge, we will follow a similar derivation to that in [63], where the authors derive a relation for the shape of an arbitrary circle projected into an image plane. Here, we will consider a restricted case where the plane of the circle is always orthogonal with the axis of rotation. Recent developments for optical navigation in the Earth-Moon system have employed a similar approach, finding that spheres, and by extension circles, project to ellipses in the image plane [14, 57, 13]. The derivations in this section will reveal that the motion of stars follows a similar geometric model.

To begin, recall the pinhole camera model illustrated in Figure 2.3. This projection assumes that a point in $\mathbb{R}^3$ is connected by a line through the focal point of the camera to its projected location on the focal plane. We will begin our derivation by considering a rotation axis aligned with the boresight of the camera. In this case, inertially-fixed objects trace circles parallel to the focal plane of the sensor. The path of the line of sight vectors describes a right cone. Figure 3.5 illustrates a two-dimensional cross-section of this geometry. The figure defines a number of parameters that will be useful later on. The opening angle of the cone is denoted by γ. This angle is equivalent to the angle between the line of sight vector of a particular object and the axis of rotation. The angle γ is always constant for a constant rotation axis. The cone extends out to infinity, but for simplicity Figure 3.5 only considers its extent out to a side length of 1 from the vertex. The side length of the cone corresponds to the length of the line of sight unit vector. So, the height of the cone is $\cos\gamma$ and the radius at the top is $\sin\gamma$. Define r and d as this radius and height, respectively.

$$r = \sin\gamma, \qquad d = \cos\gamma \tag{3.171}$$

These terms will be useful in the remainder of the derivation. Though these two variables explicitly depend on the angle, γ, we will find later that knowledge of the opening angle of the cone is not necessary for our purposes.

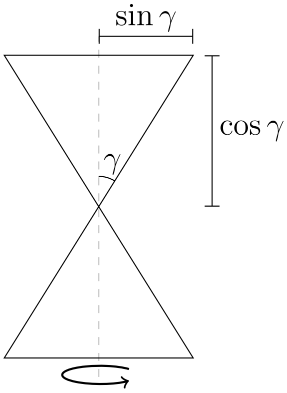

Figure 3.5: Conic geometry of a line of sight vector from the focal plane to a star.

So, the line of sight vector, $\boldsymbol{u}$, at any point in time satisfies the general form of a right cone

$$u_1^2 + u_2^2 - c^2 u_3^2 = 0 \tag{3.172}$$

where c is some constant. From Figure 3.5 we can determine a relation for c. According to the diagram, at a distance d from the vertex of the cone the circle has radius r, which is related to the opening angle, γ. So, at $u_3 = d$ the relationship between u_1 and u_2 is

$$u_1^2 + u_2^2 = r^2 \tag{3.173}$$

Inserting these values into the cone equation, we find that

$$r^2 - C^2 d^2 = 0 \quad \implies \quad C = \frac{r}{d} \tag{3.174}$$

Thus, the cone described by a line of sight vector rotating about a fixed axis of rotation is

$$u_1^2 + u_2^2 - \frac{r^2}{d^2} u_3^2 = 0 \tag{3.175}$$

This equation is useful because it describes the set of all possible line of sight vectors for a particular object. Determining the corresponding path of that object across the image plane simply requires that we evaluate the equation above at $u_3 = -f$ (the focal plane). The current geometry only considers a single case, however, so we must express the cone in a more general form to proceed. Typically the axis of rotation and the boresight will not be aligned, so the equation above does not fully describe the range of possible geometries. To fully capture the geometry of the problem, this cone must be able to account for an arbitrary orientation of the axis of rotation in the sensor's reference frame. Figure 3.6 illustrates the general case. In this figure, the axis of rotation, signified by the gray dotted line, is not orthogonal to the focal plane. Rather, it is free to exist in any orientation. Regardless of the orientation of the cone, the focal plane cuts it at a distance of f below the vertex, which corresponds to the focal point.

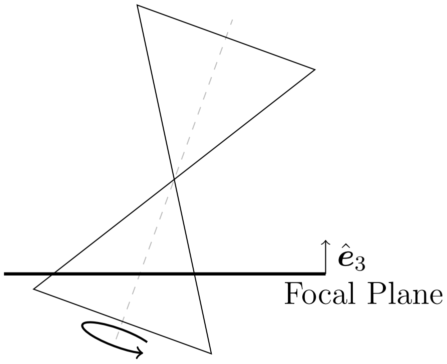

Figure 3.6: Rotated conic geometry intersecting with the focal plane.

Now let us determine a rotation matrix that will rotate the cone described by Equation 3.175 to align with an arbitrary axis of rotation. Let the axis of rotation be the vector $\boldsymbol{a}$. The vector perpendicular to both the axis of rotation and the normal to the focal plane, $\hat{\boldsymbol{e}}_3$, is

$$\boldsymbol{a}_\perp = \hat{\boldsymbol{e}}_3 \times \boldsymbol{a} \tag{3.176}$$

The angle between the axis of rotation and the focal plane normal is

$$\theta_\perp = \cos^{-1}\left(\hat{\boldsymbol{e}}_3 \cdot \boldsymbol{a}\right) = \cos^{-1}\left(a_3\right) \tag{3.177}$$

So, we can build a rotation matrix by choosing $\boldsymbol{a}_\perp$ as its rotation axis and $\theta_\perp$ as the magnitude of the rotation. From the axis-angle representation of a rotation, we can build the direction-cosine matrix. The rotation matrix, R, is [8]

$$R = I - \tilde{\boldsymbol{a}}_\perp \sin\theta_\perp + \tilde{\boldsymbol{a}}_\perp^2 \left(1 - \cos\theta_\perp\right) \tag{3.178}$$

We will follow a similar convention to the authors of [63], by partitioning the rotation matrix as follows:

$$R \equiv \begin{bmatrix} \boldsymbol{n} & \boldsymbol{o} & \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} n_1 & o_1 & a_1 \\ n_2 & o_2 & a_2 \\ n_3 & o_3 & a_3 \end{bmatrix} \tag{3.179}$$

By this convention, the transformed line of sight vectors are

$$\begin{aligned} u_1' &= u_1 n_1 + u_2 o_1 + u_3 a_1 \\ u_2' &= u_1 n_2 + u_2 o_2 + u_3 a_2 \\ u_3' &= u_1 n_3 + u_2 o_3 + u_3 a_3 \end{aligned} \tag{3.180}$$

So, now we can write the rotated cone in the following form.

$$(u_1')^2 + (u_2')^2 - \frac{r^2}{d^2} (u_3')^2 = 0 \tag{3.181}$$

Now it is possible to determine the corresponding path across the focal plane. Evaluating the Equation above at $u_3 = -f$ produces a conic section describing an object's path. Further, at $u_3 = -f$, u_1 and u_2 correspond to the focal plane coordinates x and y, respectively. So, in the general form of a conic section, we have [14]

$$p_1 x^2 + p_2 xy + p_3 y^2 + p_4 x + p_5 y + p_6 = 0 \tag{3.182}$$

Where p_i for $i = 1, \dots, 6$ are constant coefficients.

$$\begin{aligned} p_1 &= n_1^2 + n_2^2 - n_3^2 \tfrac{r^2}{d^2} & p_4 &= 2f\left(a_1 n_1 + a_2 n_2 - a_3 n_3 \tfrac{r^2}{d^2}\right) \\ p_2 &= 2\left(n_1 o_1 + n_2 o_2 - n_3 o_3 \tfrac{r^2}{d^2}\right) & p_5 &= 2f\left(a_1 o_1 + a_2 o_2 - a_3 o_3 \tfrac{r^2}{d^2}\right) \\ p_3 &= o_1^2 + o_2^2 - o_3^2 \tfrac{r^2}{d^2} & p_6 &= f^2\left(a_1^2 + a_2^2 - a_3^2 \tfrac{r^2}{d^2}\right) \end{aligned} \tag{3.183}$$

Figure 3.7 illustrates the geometry described by the equations above. The left-hand side of the figure contains a simulated, extremely wide field of view star image with a fixed axis of rotation and a long exposure. The image on the right-hand side of the figure contains a set of star paths for a fixed rotation axis, predicted by the equations above. These figures show that all of the stars rotate about a central location, with stars further from that central point follow more eccentric paths. Stars on the right-hand side of the images eventually "escape" into parabolic and hyperbolic trajectories across the image plane. These trajectories occur when the star is predicted to pass behind the sensor, where this geometric model no longer applies.

Another feature of note in the right-hand side of Figure 3.7 is that no two paths intersect. This follows intuitively from the fact that rotation should not change the positioning of stars relative to each other. So, we can infer that each (x, y) location in the image lies on a unique conic section. Indeed this is the case and it is possible to derive a unique path based on a defined starting location.

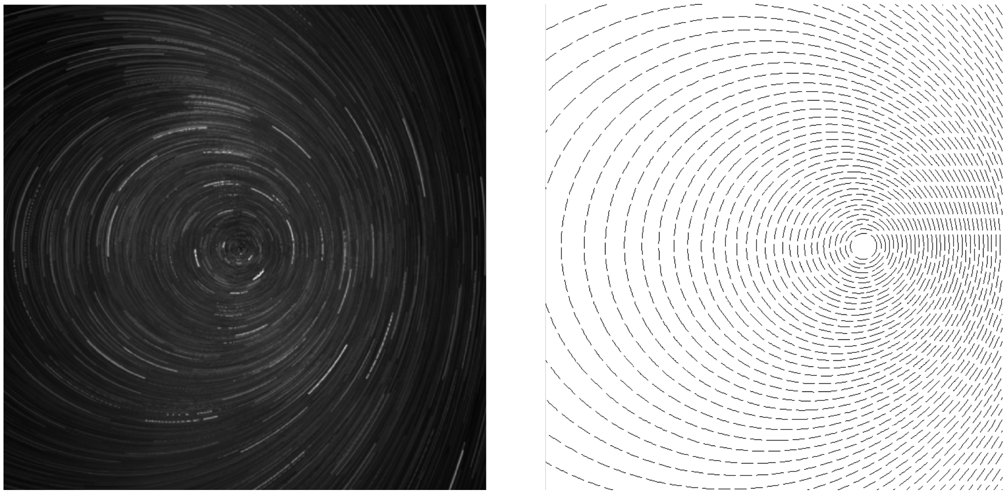

Figure 3.7: Simulated star trails (left) and geometrically-derived star paths (right).

From knowledge of the rotation axis, the only unknown term is r^2/d^2. For a particular (x, y) point in the image plane it is possible to solve directly for this relationship, thus providing a unique path description for that starting location. Rearranging Equation 3.182 to solve for r^2/d^2 gives

$$\frac{r^2}{d^2} = \frac{k_1 x^2 + k_2 xy + k_3 y^2 + k_4 x + k_5 y + k_6}{\left(n_3 x + o_3 y + a_3 f\right)^2} \tag{3.184}$$

where k_i for $i = 1, \ldots, 6$ is

$$\begin{aligned} k_1 &= n_1^2 + n_y^2 & \qquad k_4 &= 2f\left(a_1 n_1 + a_2 n_2\right) \\ k_2 &= 2\left(n_1 o_1 + n_2 o_2\right) & \qquad k_5 &= 2f\left(a_1 o_1 + a_2 o_2\right) \\ k_3 &= o_1^2 + o_2^2 & \qquad k_6 &= f^2\left(a_1^2 + a_2^2\right) \end{aligned} \tag{3.185}$$

And thus we are able to solve for a value of $\frac{r^2}{d^2}$ for any initial (x, y) coordinate from Equation 3.184. Inserting the result back into Equation 3.182 yields a unique path across the image plane. With no variation in the axis of rotation, each object stays on a single path for all time, completing one cycle each time the sensor completes a full rotation.

There are three possible types of conic sections which may arise from Equation 3.182. From the parameters, p_i, these conic sections are

$$\begin{aligned} p_2^2 - 4p_1p_3 &< 0 & \qquad &\Longrightarrow & \qquad &\text{Ellipse} \\ p_2^2 - 4p_1p_3 &= 0 & \qquad &\Longrightarrow & \qquad &\text{Parabola} \\ p_2^2 - 4p_1p_3 &> 0 & \qquad &\Longrightarrow & \qquad &\text{Hyperbola} \end{aligned} \tag{3.186}$$

Conveniently, there are some degenerate cases for general conic sections that need not be considered here. Namely, the case where the solution to Equation 3.182 is a point, a single line, or two intersecting lines. Since the cutting plane does not pass through the focal point (because the focal length is strictly greater than zero), these cases will never occur [64].

Note that this result – that stars travel along conic in the image plane – coincides with the constant rotation observer results derived earlier. Equation 3.75 and the derivation leading up to it showed that stars revolve around the center of rotation in paths that approximate circles for a small field of view. We now see that the true path is described by a conic section, which may be approximated as circular for small segments. Interestingly, Figure 3.7 shows the location of the center of rotation from Equation 3.67. Approaching the center of rotation, stars follow smaller, more circular paths.

There are a number of ways to leverage this knowledge to produce robust, generalized space object detection routines. Chapter 5 will revisit this work and explore some of its applications.

3.4 Multi-Observer Epipolar Geometry

Thus far we have only considered single-sensor systems. Though this is the most common type of system in use, there are some benefits to analyzing groups of sensors simultaneously. This section explores the geometry arising from multi-sensor systems. Optical space situational awareness systems typically rely on several disparate sensors operating independently but tasked according to current tracking needs [65]. There have been some recent efforts to leverage multi-sensor systems for the SSA problem. In [66] and [67], the authors demonstrate the use of two synchronized sensors to make fast space object detections and with simultaneous range measurements. Chapter 5 will approach the same problem from a more general standpoint – developing a technique of detection and ranging space objects with any number of sensors.

The methods derived in this dissertation rely on a concept called epipolar geometry. The term epipolar geometry describes the relative vantage points of two optical sensors. The basic principle behind stereo vision (a dual-sensor system) leverages the inter-sensor geometry to immediately determine the position of an object of interest. The range of an object is not typically available to a single sensor, but a second vantage point can reveal the object's range [19]. The key challenge of this sensor setup involves associating objects across sensors. This challenge is amplified in the SSA problem because objects are typically unresolved and have no identifying features. Epipolar geometry offers a solution to this problem.

Figure 3.8 illustrates the way that this section will use epipolar geometry. In this figure, a line of sight vector measured by a secondary sensor passes above the primary sensor. This line of sight vector indicates the relative position of an object to the secondary sensor. Though the secondary sensor is not able to measure the range to the object, we do know that the object of interest lies at some distance along that line. Further, since the object lies along that line we infer that the line of sight, when projected into the primary sensor, will intersect with the object's projected location in that sensor. Based on the geometry it is apparent that a line in $\mathbb{R}^3$ projects to a line in $\mathbb{R}^2$ in the pinhole camera model.

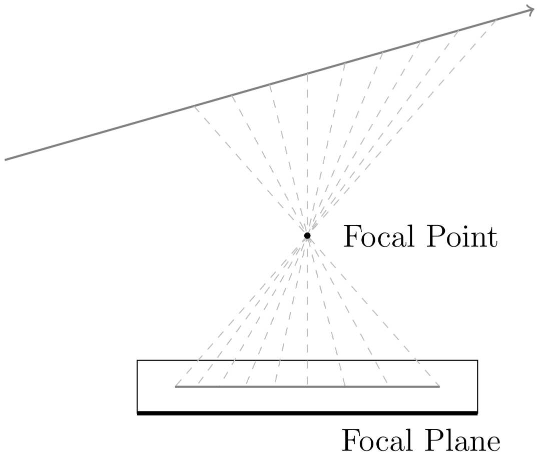

Figure 3.8: A line of sight vector projecting to a line in a secondary sensor.

So, we are interested in expressing a line of sight vector from a secondary observer in the primary sensor. Begin by defining the line of sight vector according to

$$\boldsymbol{r}_u = \boldsymbol{r}_o + \ell \boldsymbol{u} \tag{3.187}$$

where $\boldsymbol{r}_o$ is the position of the secondary observer in the primary sensor's body-fixed reference frame and $\boldsymbol{u}$ is the unit vector describing the optical measurement of an object of interest from that sensor. The line of sight vector, $\boldsymbol{u}$, is also expressed in the reference frame of the primary sensor. From the pinhole projection model (Equation 2.4), the image plane x coordinate corresponding to any point on the line of sight vector is

$$x = -f \frac{(\boldsymbol{r}_o)_1 + \ell u_1}{(\boldsymbol{r}_o)_3 + \ell u_3} \tag{3.188}$$

where $(\cdot)_i$ indicates the ith element of the vector in the parentheses. Similarly, projecting the line of sight vector through the Collinearity equation yields a y coordinate of

$$y = -f \frac{\ell u_2 + (\boldsymbol{r}_o)_2}{\ell u_3 + (\boldsymbol{r}_o)_3} \tag{3.189}$$

We desire a relationship for this line expressed entirely in the image plane. Now, solving the x equation for ℓ yields

$$\ell = \frac{-f\,(\boldsymbol{r}_o)_1 - x\,(\boldsymbol{r}_o)_3}{f u_1 + x u_3} \tag{3.190}$$

and solving the y equation in the same way shows that

$$\ell = \frac{-f\,(\boldsymbol{r}_o)_2 - y\,(\boldsymbol{r}_o)_3}{f u_2 + y u_3} \tag{3.191}$$

So, from these two relations it is possible to derive an equation fully in terms of x and y. Setting both equations for ℓ equal to each other gives

$$\frac{-f\left(\boldsymbol{r}_o\right)_1 - x\left(\boldsymbol{r}_o\right)_3}{fu_1 + xu_3} = \frac{-f\left(\boldsymbol{r}_o\right)_2 - y\left(\boldsymbol{r}_o\right)_3}{fu_2 + yu_3} \tag{3.192}$$

which, when solved for y in terms of x, is of the form

$$y = mx + b \tag{3.193}$$

where m describes the slope of the line and b is the y-intercept. The slope of this line is

$$m = \frac{u_3\left(\boldsymbol{r}_o\right)_2 - u_2\left(\boldsymbol{r}_o\right)_3}{u_3\left(\boldsymbol{r}_o\right)_1 - u_1\left(\boldsymbol{r}_o\right)_3} \tag{3.194}$$

and the y-intercept is

$$b = f\frac{u_2\left(\boldsymbol{r}_o\right)_1 - u_1\left(\boldsymbol{r}_o\right)_2}{u_1\left(\boldsymbol{r}_o\right)_3 - u_3\left(\boldsymbol{r}_o\right)_1} \tag{3.195}$$

Thus we have proved that a line in $\mathbb{R}^3$ projects to a line in $\mathbb{R}^2$ under the assumptions of the pinhole camera model. Further, the resulting line in the image plane is fully defined by knowledge of the focal length, f, the line of sight unit vector, $\boldsymbol{u}$, and the secondary sensor's position in the primary sensor's reference frame, $\boldsymbol{r}_o$.

An alternative representation of this line is possible through the normal form parameterization of a line. The normal form defines a line by its orthogonal distance form the origin and the angle its bisector makes with the x axis. Figure 3.9 illustrates the normal form parameters for a line [4]

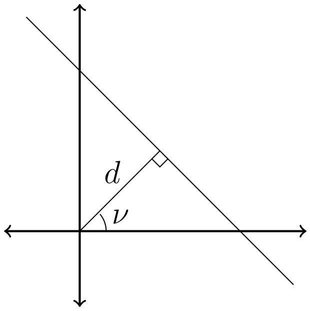

Figure 3.9: Normal form line parameterization [4].

The standard equation for a line represented in the normal form is

$$d = x\cos\nu + y\sin\nu \tag{3.196}$$

This formulation has the benefit of removing the singularity for vertical lines. Though the case where the geometry aligns in such a way that it causes a perfectly vertical line is unlikely, this form can reduce numerical issues arising from an excessively large slope. The relationships between the slope, m, and the y-intercept, b, and the parameters of the normal form follow from examination of Equation 3.196. The angle, ν, is

$$\nu = \tan^{-1}\left(-\frac{1}{m}\right) = -\cot^{-1}(m) \tag{3.197}$$

and the distance from the origin, d, is

$$d = b \sin \nu \tag{3.198}$$

3.5 Summary

The work outlined in this chapter makes up the theoretical basis of the proposed work in the remaining chapters. These fundamental motion models are novel representations of the motion of objects viewed through an optical sensor. The models attempt to capture the expected motion of a space object across the image plane of a sensor to better prepare data processing algorithms for input from a diverse set of optical observers.

It is important to remember that the derivations here are based on a number of different assumptions. Each section works through the assumptions for the approximate solutions, but the key assumption shared by all of these solutions is that the optical sensor follows a known model (either the pinhole or distortion-corrected model) and that the sensor has a constant angular rotation rate. So all of these models are vulnerable to improper modeling of the sensor and angular accelerations. If the sensor does not follow the assumed model, then this analysis will break down and provide erroneous predictions. In particular, the distortion-corrected motion models are particularly sensitive to the accuracy with which the distortion correction is applied to the original image data. It is possible that regions of the image deviate locally from the assumed model, causing variable performance over the focal plane.

We will explore the Equations of motion contained in Sections 3.1 and 3.2 in the following chapter. These representations of the dynamics of space objects in the reference frame of an optical sensor enable a number of sensor-level detection and estimation routines to reduce optical data in a more general way. As we saw in Section 2.1, typical data reduction approaches are designed for a particular sensor in a particular operating mode. The work in the coming chapters will address this fact in an effort to enable a centralized data processing pipeline. Chapter 5 will examine more closely the implications of the geometric representation from Sections 3.3. We will show that this description of the motion allows for an alternative set of analysis that are capable of correcting for non-ideal sensor tasking.

Chapter 4

Implications of the Dynamic Analysis

This chapter will leverage the results from the previous chapter to derive general detection, tracking, and classification algorithms for unresolved space imagery.

4.1 Space Object Analysis

This section will consider the use of the dynamic equations of motion from Chapter 3 for unresolved space object tracking and classification. Specifically, the linear dynamics from Equations 3.60 and 3.106 enable a Kalman filter based approach to tracking unresolved space objects in real time. Further, we can leverage these same motion models to stochastically determine which signals in an image frame are stars and which are objects of interest. We will also find that knowledge of the motion of objects in the image frame can aid in object extraction and star removal. Some of the algorithms described in this section have enabled a MATLAB-based toolbox for processing general space imagery from diverse sources. For more on this software package, see [68].

4.1.1 Frame-to-Frame Tracking

Frame-to-frame point tracking in sequential imagery may be achieved in a number of ways. The most commonly cited approaches focus on resolved imagery. The Scale-Invariant Feature Transform, for example, reduces an image to a set of unique feature points. The algorithm describes each of these feature points in a precise way that can then be compared to features in other images [69]. A number of extensions to this algorithm exist that enhance its speed [70] and robustness [71]. Since these techniques rely on the ability to uniquely identify an object based on its appearance in an image, however, they are not applicable to the unresolved imagery that is typical to observations of space objects. An alternative method, the Kanade-Lucas-Tomasi (KLT) tracking routine, relies somewhat on the appearance of an object in the image, but uses optical flow to derive frame-to-frame associations. The KLT tracker extracts a small window of the data and compares that window frame-to-frame. A minimization problem then calculates the shift between frames by comparing the window to a neighboring area [72].

The literature in point tracking without identifiable features commonly focuses on computing the optimal frame-to-frame associations between points in consecutive frames. Many of these methods make an assumption of small, smooth movements of points between frames. Early work considered a multiple hypothesis approach to the association problem. For each image, the algorithm in [73] matches measurements with all of its possible matches, creating a new hypothesis where necessary. This algorithm continues to branch out as new measurements come in and prunes outdated tracks that are no longer likely to be correct. The work in [60] achieves measurement assocations through a “proximal uniformity” constraint which assumes that most objects move slowly along smooth paths. A proximal uniformity cost function provides a direct method of choosing the frame-to-frame object association that creates the most smooth trajectory across the image. In [61], the authors again assume slow movement along a smooth path but expands the algorithm to account for the possibility of missed measurements. Their technique allows for a maximum number of missed measurements (“maximal absence”) and a minimum number of detections (“minimal presence”). Finally, the algorithm discussed in [74] uses an a-contrario approach to estimate the likelihood of a particular trajectory appearing in random data and associates objects in a manner that reduces the number of false alarms. This algorithm constrains the acceleration of an object but not the velocity of the object, preferring only associations that describe a smooth path.

Though these tracking routines offer precise solutions for the frame-to-frame association process that are robust to missed detections, they tend to have a high computational cost and restrictive constraints. The multiple hypothesis method requires a large number of tracks which can be costly to operate on [73]. Similarly, the a-contrario approach in [74] has a high computational cost that grows rapidly as it accounts for a larger number of missed detections. Further, the methods in [60] and [61] require that the objects move slowly – a requirement that is often violated by imaged space objects. For tracking space objects this dissertation relies on a dynamically-inspired estimation approach. Understanding the dynamics of the objects in view informs the process of their expected behavior and opens up the data reduction algorithm to a wide variety of sensor motion profiles. This work will be more in line with the work in [75], where the authors examine different multiple-hypothesis filtering approaches to track objects in an image. In particular, we will consider a Kalman filter bank built on the motion models from Chapter 3. It is important to note, however, that other point tracking routines may provide benefits in different areas. The work here aims to reduce the computational cost for on-line data processing but if computational cost is not a concern than an optimal data association process may provide better tracking performance.

There are many different possible approaches to tracking space objects across the image plane based on the motion models derived in Chapter 3. For brevity we will only fully consider a single approach based on the approximate, linear pinhole motion models. This implementation relies on a bank of multiple-hypothesis Kalman filters (MHKFs) to track each object in view across the image plane. Each MHKF is built on the analysis from Chapter 3 and accommodates multiple potential motion profiles and ultimately converge on the model with the best fit. Specifically, the MHKF uses the model for the rotating observer with inertially-fixed objects to track stars and the model for a nonrotating sensor with a constant velocity object to track non-star space objects. Though the translational motion model is an approximation of the true motion of a non-star object across the image plane we will find that, in practice, this model typically provides accurate tracking performance.

Recall that the approximate equation of motion describing the path of an inertially-fixed object across the focal plane of a sensor rotating with constant angular rate is

$$\left[\begin{array}{c} x\left(t\right) \\ y\left(t\right) \end{array}\right] = \left[\begin{array}{cc} \cos\left(\omega_3 t\right) & \sin\left(\omega_3 t\right) \\ -\sin\left(\omega_3 t\right) & \cos\left(\omega_3 t\right) \end{array}\right] \left[\begin{array}{c} x\left(t_0\right) \\ y\left(t_0\right) \end{array}\right] + \frac{f}{\omega_3} \left[\begin{array}{cc} \cos\left(\omega_3 t\right) - 1 & \sin\left(\omega_3 t\right) \\ -\sin\left(\omega_3 t\right) & \cos\left(\omega_3 t\right) - 1 \end{array}\right] \left[\begin{array}{c} \omega_1 \\ \omega_2 \end{array}\right] \tag{4.1}$$

Note that this equation requires knowledge of the angular rate vector, $\boldsymbol{\omega}$. The inertially-fixed object motion hypothesis must be initiated with an estimate of the angular rate of the sensor. If this is not immediately available, it is possible to estimate the angular rate vector from the image data. There are two primary methods for solving for the angular rate vector: through star identification or from a frame-to-frame rotation estimate. Both approaches require at least 3 stars in view, though additional stars increase the robustness of either method. An angular rate estimate from either method is limited by the fact that we only sample the attitude at discrete points in time which are sometimes separated by several seconds. Any acceleration occurring over that time period will be ignored. In general, however, these estimates provide a reasonable initial guess.

Star identification techniques like the small-scale Pyramid method or the large-scale Astrometry.net tools provide precise sensor attitude estimates based on analysis of the background star field [16, 18]. These methods compare geometric relationships describing the relative locations of stars in an image to a known catalog of stars. Though this process can be computationally expensive for a large star catalog, it typically provides highly accurate catalog matches [17]. Then, once a set of has been matched to the corresponding catalog entries, it is possible to infer the rotation of the observed stars from the reference stars. This rotation describes the attitude of the sensor. The problem of estimating the rotation between two sets of vectors is known as Wahba's problem. Wahba's problem is formulated as an optimization problem with the cost function [76]

$$L\left(\hat{R}\right) = \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{b}_i - \hat{R}\boldsymbol{r}_i\right\|_2^2 \tag{4.2}$$

where $\boldsymbol{b}_i$ represents the line of sight vector corresponding to the ith observed star and $\boldsymbol{r}_i$ represents the line of sight vector corresponding to the ith reference star. Figure 4.1 illustrates this problem graphically. There are numerous computationally efficient, optimal solutions to this problem [77]. In order to solve this problem, we require vectors in $\mathbb{R}^3$ rather than just the image-plane coordinates. Recall the inverse projection described in Equation 2.6. This equations gives a unit norm line of sight vector corresponding to an (x, y) location in the image plane. From that set of vectors and the line of sight vectors describing the reference stars in the catalog it is then possible to solve for an attitude estimate.

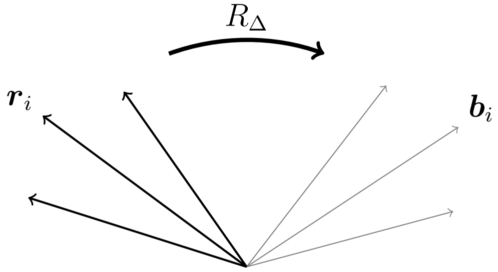

Figure 4.1: Illustration of Wahba's problem (R_Δ is unknown).

Now assume that the star identification procedure succeeds for two successive images. For each frame we have a corresponding sensor attitude derived from the background star field. Denote the direction cosine matrix (DCM) describing the initial attitude estimate by $\hat{R}_i$ and the DCM describing the attitude corresponding to the second image as $\hat{R}_f$. We can then infer that the direct rotation from the initial attitude to the final attitude is

$$\hat{R}_\Delta = \hat{R}_i \hat{R}_f^T \tag{4.3}$$

It is from this rotation matrix that we will derive the angular rate vector of the spacecraft. Assuming that the angular rate was constant (or at least approximately constant) between the two frames, we can estimate the angular rate by considering the axis-angle representation of $\hat{R}_\Delta$. If the sensor rotated with a constant rate, then the angular rate follows from the magnitude of the rotation and the time over which that rotation occurred. The axis about which the rotation described by $\hat{R}_\Delta$ revolves is [8]

$$\tilde{\boldsymbol{a}} = \frac{1}{2\sin\Phi}\left(\hat{R}_\Delta - \hat{R}_\Delta^T\right) \tag{4.4}$$

where $\tilde{\boldsymbol{a}}$ describes the skew-symmetric form of the axis of rotation vector, $\boldsymbol{a}$. The angular magnitude of displacement around $\boldsymbol{a}$ is [8]

$$\Phi = \cos^{-1}\left\{\frac{1}{2}\left(\operatorname{tr}\left(\hat{R}_\Delta\right) - 1\right)\right\} \tag{4.5}$$

So then the skew-symmetric form of the angular rate vector is

$$\tilde{\boldsymbol{\omega}} = \frac{\Phi}{\Delta t}\tilde{\boldsymbol{a}} \tag{4.6}$$

where Δt is the time between frames. Similar approaches have been developed for estimation of the angular rate of the sensor from a single image frame [49, 46]. These methods rely on the stars to be imaged in such a way as to cause significant movement over the course of the exposure time. This assumption removes some of the generality of the overall approach and thus will not be considered here.

Now, recall that the approximate equation of motion describing the path of a non-star object across the image plane of a nonrotating sensor was shown in Equation 3.106 to be

$$\begin{bmatrix} x\left(t\right) \\ y\left(t\right) \\ \dot{x}\left(t\right) \\ \dot{y}\left(t\right) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x\left(t_0\right) \\ y\left(t_0\right) \\ \dot{x}\left(t_0\right) \\ \dot{y}\left(t_0\right) \end{bmatrix} \tag{4.7}$$

Though this model does not account for the rotation of the sensor, it is still typically able to provide accurate tracking performance in practice. Ideally, this solution would be swapped out for the solution in Equation 3.93 or 3.94. Even though these equations do not consider the angular rate of the sensor, it is still possible to derive a velocity estimate in order to initialize the Kalman filter. If an object of interest is moving quickly enough that it has some extent in the image (i.e. it is streaked) and we are able to extract precise endpoint measurements from that signal, then it is possible to estimate its image-plane velocity. The extraction of endpoints of streaks is non-trivial, though there are some solutions currently. In [78], the authors develop an iterative fit to approximate the signal near the endpoints, achieving accuracies of approximately a tenth of a pixel. One may also leverage a gradient-based image analysis method to extract the endpoints. Specifically, the geometry of a streaked object is such that a corner detector produces its peak response at the endpoints of the signal. See [46] for more information on this technique.

Assume that we are able to measure the endpoints of a streaked point source with some mean, $\bar{\boldsymbol{e}}$, and uncertainty, Σ_e.

$$\boldsymbol{e}_i \sim N\left(\bar{\boldsymbol{e}}_i, \Sigma_e\right) \tag{4.8}$$

Computation of a measurement covariance for the endpoints is also non-trivial. There is some discussion of this in the literature for feature tracking, but no conclusive results [79, 80, 81]. A rudimentary option is computation of the second derivative (the Hessian) of the intensity at that point. This approach operates under the assumption that signals which are smaller in the image are going to be more precisely localized [82]. In practice, this method tends to be a conservative solution and often overshoots the true uncertainty.

If the object moves with a constant velocity, as is assumed in Equation 4.7, then the velocity estimate arising from measurements of the two endpoints, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, is

$$\dot{\boldsymbol{x}} = \frac{\boldsymbol{e}_2 - \boldsymbol{e}_1}{\Delta t} \tag{4.9}$$

where Δt is the integration time of the sensor and $\boldsymbol{x} = (x, y)$. So, based on the uncertainty of the two endpoint measurements, the resulting distribution of the velocity estimate is

$$\hat{\dot{\boldsymbol{x}}} \sim N\left(\frac{\bar{\boldsymbol{e}}_2 - \bar{\boldsymbol{e}}_1}{\Delta t}, \frac{2}{\Delta t^2}\Sigma_e\right) \tag{4.10}$$

In this case, we assume that $\boldsymbol{e}_2$ is the endpoint corresponding to the end of the exposure time. In general, we do not know which endpoint occurred first or second. To resolve this ambiguity, we will consider a hypothesis for both the positive and negative velocity cases: $\pm\hat{\dot{\boldsymbol{x}}}$. Following a second measurement, this ambiguity should be removed and only one hypothesis will remain valid.

Now, based on the two motion models above, we can construct the MHKF filter bank. Assume that the systems in Equations 4.1 and 4.7 are of the form [83]

$$\boldsymbol{x}_{k+1} = \Phi_k \boldsymbol{x}_k + \Gamma_k \boldsymbol{u}_k + \Upsilon \boldsymbol{w}_k \tag{4.11}$$

where $\boldsymbol{w}_k$ is normally-distributed process noise. The measurement model is of the form

$$\boldsymbol{z}_k = H_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{4.12}$$

where $\boldsymbol{v}_k$ is normally-distributed measurement noise. For both motion models, the measurement is the position of the object in the focal plane. We saw previously that it is possible to estimate the velocity of an object, so it is possible to formulate the Kalman filters for the motion model in Equation 4.7 to also accept a velocity measurement, but that is not the case in this particular implementation. Recall from Table 2.2.3 that the prediction equations for the mean state, $\hat{\boldsymbol{x}}_{k+1}^-$, and the covariance, P_{k+1}^+, are

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^- &= \Phi \hat{\boldsymbol{x}}_k^+ + \Gamma \boldsymbol{u}_k \\ P_{k+1}^- &= \Phi_k P_k^+ \Phi_k^T + \Upsilon Q_k \Upsilon_k^T \end{aligned} \tag{4.13}$$

The $^-$ superscript indicates that these values have been predicted forward without a correction step in the presence of a new measurement. The Kalman gain is [83]

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \tag{4.14}$$

Finally, from a measurement, $\boldsymbol{z}_k$, and the Kalman Gain K_k, the corrected state mean and covariance is

$$\begin{aligned} \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + K_k\left(\boldsymbol{z}_k - H_k \hat{\boldsymbol{x}}_k^-\right) \\ P_k^+ &= \left(I - K_k H_k\right) P_k^- \end{aligned} \tag{4.15}$$

The Kalman filter process is well-known and simple the implement. Perhaps the most significant challenge of this MHKF filter bank is the frame-to-frame association process. There are a number of ways to approach the object association problem and the performance of any method will be closely coupled to the density of objects in the image.

For this particular implementation the Global Nearest Neighbor (GNN) technique provides the proper associations between predicted tracks and measurements. This method works by associating tracks with the nearest measurement with the additional constraint that no measurement may be associated to multiple tracks [84]. If $\boldsymbol{m}_j \in \mathbb{R}^2$ is a measurement and $\boldsymbol{x}_{k+1}^{[i]-}$ is the prediction for the ith tracked object based on prior knowledge, the correct match is the one that satisfies

$$\min_i \left\{ \left\| \boldsymbol{x}_{k+1}^{[i]-} - \boldsymbol{m}_j \right\| \right\} \tag{4.16}$$

Further, this measurement satisfying the equation above must also pass through a gating procedure. Here, we constrain associations to only those that are within σ_{limit} of track prediction where σ_{limit} is an upper limit based on the current uncertainty of the track in questions. The distance between a predicted track and a point, $\boldsymbol{p} \in \mathbb{R}^2$, is given by the Mahalanobis distance [85]:

$$d\left(\boldsymbol{p}\right) = \sqrt{\left(\boldsymbol{p} - \boldsymbol{x}_{k+1}^{-}\right)^T P^{-1} \left(\boldsymbol{p} - \boldsymbol{x}_{k+1}^{-}\right)} \tag{4.17}$$

So, a positive match between a predicted track and a measurement, $\boldsymbol{m}_i$, satisfies Equation 4.16 and

$$d\left(\boldsymbol{m}_i\right) < \sigma_{\text{limit}} \tag{4.18}$$

If a measurement satisfies Equation 4.16 for multiple tracks, then a second step is necessary to ensure that the measurement is only matched with one track. In that case, the measurement matches with the nearest track prediction and all other tracks remain unassociated. Alternatively, it is possible to incorporate multiple measurements into the prediction step for each track through the Joint Probabilistic Data Association (JPDA) method. JPDA updates each track with every measurement falling within σ_{limit}, weighted according to its distance from the predicted location [86]. This process avoids making hard decisions when there is a possible ambiguity. A caveat of this approach, though, is that closely-space objects have a tendency to coalesce [87].

It is worth noting that a star identification process, if successful, can provide some insight into the correct frame-to-frame object associations [16, 18]. For example, the Astrometry.net software package is capable of matching stars in an image to those in its catalog through analysis of the inter-star geometry in an image. The result provides a unique identifier for each star [18]. If the star was identified in a previous frame as well, then the correct association is immediately available. It is important to note that no star identification package guarantees a solution. In fact, while Astrometry.net has a high success rate in general, it will fail for a low number of visible stars (< 4). Further, the computational cost of searching a large star catalog can be prohibitive for real-time applications. For these reasons, we approach this problem as if the correct frame-to-frame associations are not available. If a star identification process is available, then it can augment the solution proposed here.

There are several ways to prune outdated tracks from the track database. All of the techniques considered here rely on analysis of the position and position covariance of each track.

The first method is the most basic and relies on the predicted location of the object in the next image. If the object is predicted to pass outside of the bounds of the image, then it should be removed from the database. This constraint prevents a large number of tracks from accumulating as stars and space objects move through the frame.

Another consideration for removing unnecessary tracks is the current position uncertainty. A useful parameter for determining the appropriate choice for the uncertainty bound is a measure of the density of objects in the image. Define the "field sparsity" parameter as

$$\rho_f = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n}\|\boldsymbol{m}_i - \boldsymbol{m}_j\|\right) \tag{4.19}$$

where $\boldsymbol{m}_i$ is the location of the ith object measured in a single frame. In words, this is half of the average distance between objects in the image. We refer to this parameter as the field sparsity, rather than the density, because the average distance between objects grows as the number of objects in the field decreases. So a more sparse image will return a larger field sparsity parameter and a more densely packed image will return a lower value for ρ_f. This number is useful because it provides a measure of the likelihood of a conflict between track predictions. If the image is sparse, then we can allow a larger position uncertainty for any tracked object. If the image is more dense, then a large position uncertainty may lead to incorrect measurement associations.

This tracking routine has been implemented in a MATLAB toolbox targeted at reducing sets of sequential imagery from diverse observers with limited sensor knowledge. For more information on this software package, see [68]. Though this particular implementation uses the linear approximate models from Chapter 3, any of the models in that Chapter may be implemented in a similar fashion. In practice there are cases that violate the small field of view and pinhole optics assumptions. When this happens, the tracks for both valid stars and non-stars tend to drop due to the unmodeled motion. In those situations, a higher fidelity motion model may provide improved tracking performance.

4.1.2 Object Classification

Now, the tracking results alone do not necessarily reveal the correct classification of objects as stars or non-star objects of interest. Though the fact that a track might prune the star motion model (Equation 4.1) from its filter process suggests that the corresponding object may not be a star, it is still uncertain. In fact, when the observer rotates about a single axis, we can see from the paths in Figure 3.2 that the simple translation model captures the motion of a star very closely for a small field of view. Thus we require a secondary process to determine if a track does in fact correspond to a non-star space object.

One possible technique for discriminating between stars and non-star objects of interest is to consider the rotation of the sensor frame-to-frame. We have seen that Wahba's problem allows us to estimate the rotation between two sets of vector observations. It is possible to formulate Wahba's problem in such a way as to determine only the rotation between sequential image frames. As long as there are at least three stars in an image and they make up the majority of the visible signals (i.e. stars largely outnumber the space objects in view) then we can derive an estimate for the rotation. Consider a formulation of Wahba's problem slightly different from that in Equation 4.2.

$$L\left(\hat{R}_\Delta\right) = \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{u}_i^- - \hat{R}_\Delta \boldsymbol{u}_i^+\right\|_2^2 \tag{4.20}$$

Here, $\boldsymbol{u}_i^-$ is the set of line of sight vectors for objects observed in the first frame, and $\boldsymbol{u}_i^+$ is the set of vectors extracted from the second frame. The inverse projection from Equation 2.6 provide the vectors $\boldsymbol{u}_i^-$ and $\boldsymbol{u}_i^+$ from the set of the observed object's image plane coordinates in the corresponding image. The key result from this process is not the rotation estimate, but rather the residual of the optimal solution. The residual for Equation 4.20 is

$$\epsilon = \left\|\boldsymbol{u}_i^- - \hat{R}_\Delta \boldsymbol{u}_i^+\right\|_2 \tag{4.21}$$

If an object is not moving in the same fashion as the background stars, then we would expect it to have a larger residual in this process. Therefore it is possible to leverage this estimate of the frame-to-frame rotation to indicate which objects exhibit some apparent motion and are likely not stars. Figure 4.2 contains an example of a simulated case. The image on the left contains a single non-star object against the celestial sphere. That object exhibits some of its own motion, which is apparent in the slope of the resulting streak. On the right of the figure is a plot of the residuals of this process. It is clear that there is a single outlier in the data set, and that outlier corresponds to the slanted streak on the left.

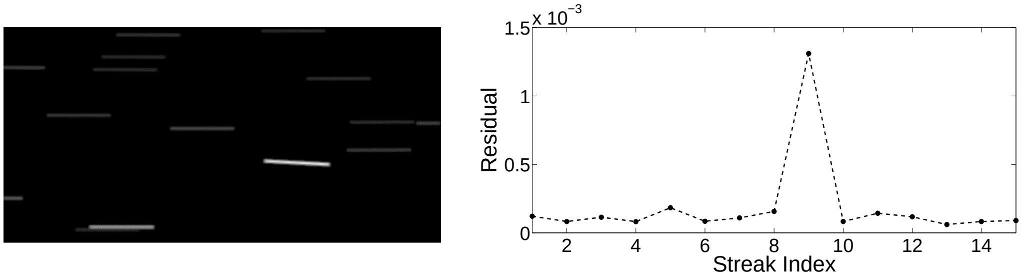

Figure 4.2: Simulated RSO image (left) and the corresponding rotation residuals [5].

An important aspect of this technique to note is that it can be unreliable for images which do not contain many stars. In addition, noise in the sensor has the potential to cause false-positive returns. A more robust algorithm to handle the tagging of non-star objects includes a voting routine, which increases an object's "RSO score" for each time that it returns an outlier. After a sufficient number of votes it should gradually become more apparent if an object is actually an RSO.

Another approach to classifying objects based on tracking data is to once again leverage the motion models from Chapter 3. As discussed earlier, just because an object is not tracked in the filter bank by the star motion model from Equation 4.1 does not necessarily mean that it is not a star. Measurement noise may lead to a case where the model in Equation 4.7 better describes the motion of a star initially, causing all other hypotheses to be pruned. We can return to the motion model for stars to determine how much a particular object deviates from that model. If it follows that model closely, then we can conclude it is a star, otherwise it is a non-star object of interest. Consider the measurement model

$$\boldsymbol{z}_k = \boldsymbol{x}_k + \boldsymbol{v}_k \tag{4.22}$$

This model provides a direct measurement of the (x, y) position of an object with some normally-distributed measurement noise, $\boldsymbol{v}_k$. Now, write Equation 4.1 in the form

$$\boldsymbol{x}_{k+1} = f_d\left(\boldsymbol{x}_k, \boldsymbol{\omega}_k, f_k, \Delta t_k\right) \tag{4.23}$$

where $\boldsymbol{\omega} = \left(\omega_1, \omega_2, \omega_3\right)^T$, f_k is the focal length at the kth time step, and Δt_k is the duration of the kth time step. For generality, we include the k subscript to indicate the value of the subscripted variable at that time step. Though we assumed that the angular rate was constant for the solution in Equation 4.1 we allow for the possibility of an instantaneous change in the angular rate between frames. In addition, the focal length is generally constant, but we will consider the possibility of small temporal variations in the optics. Also define a residual vector according to

$$\boldsymbol{w}_{k+1} = \boldsymbol{z}_{k+1} - f_d\left(\boldsymbol{z}_k, \boldsymbol{\omega}_k, f_k, \Delta t_k\right) \tag{4.24}$$

The quantity $\boldsymbol{w}_{k+1}$ of an object's motion from Equation 4.1 during the kth time step. We can also write this more compactly by defining the augmented measurement vector, $\boldsymbol{\chi}_k$,

$$\boldsymbol{\chi}_k = \begin{bmatrix} \boldsymbol{z}_k^T & \boldsymbol{\omega}_k^T & f_k & \Delta t_k \end{bmatrix}^T \tag{4.25}$$

which is treated as a Gaussian random variable with covariance

$$\Sigma_{\boldsymbol{\chi}} = \operatorname{diag}\left(\Sigma_z, \Sigma_\omega, \Sigma_f, \Sigma_t\right) \tag{4.26}$$

where Σ_z is the measurement uncertainty, Σ_ω is the angular rate uncertainty, Σ_f is uncertainty in the focal length, and Σ_t is the uncertainty in the time step. Then, the measurement residual becomes

$$\boldsymbol{w}_{k+1} = \boldsymbol{z}_{k+1} - f_d\left(\boldsymbol{\chi}_k\right) \tag{4.27}$$

This residual provides an indicator for the correct classification of an object of interest. If the object is a star, then this residual should remain small. If the object is a non-star, then its residual will gradually increase in magnitude with time as it deviates from Equation 4.1. To build a hypothesis test based on this residual, we must first determine the distribution of $\boldsymbol{w}_{k+1}$. In general, this distribution is not precisely Gaussian, but we can approximate it through a linearization process. The Jacobian of $f_d(\cdot)$ with respect to $\boldsymbol{\chi}_k$ is [83]

$$J_k = \left.\frac{\partial f_d}{\partial \boldsymbol{\chi}}\right|_{\boldsymbol{\chi}_k} \tag{4.28}$$

So, the covariance of $\boldsymbol{w}_{k+1}$ is approximately

$$\Sigma_{w_{k+1}} = \Sigma_z + J_k \Sigma_{\boldsymbol{\chi}} J_k^T \tag{4.29}$$

A single residual will likely not contain enough information to confidently classify a space object. So rather than looking at a single residual, we will consider an accumulation of the residuals over time. Define $\bar{\boldsymbol{w}}_k$ as the sum of the residuals from the first time step to the current one. Then, $\bar{\boldsymbol{w}}_k$ is

$$\bar{\boldsymbol{w}}_k = \sum_{i=1}^{k} \boldsymbol{w}_i, \qquad \Sigma_{\bar{w}_k} = \sum_{i=1}^{k} \Sigma_{w_i} \tag{4.30}$$

where

$$\bar{\boldsymbol{w}}_k \sim N\left(0, \Sigma_{\bar{w}_k}\right) \tag{4.31}$$

Thus we can build a hypothesis test based on the distribution of the residual to determine if a particular object is a star or a nonstar object of interest. It is worth noting that there are some cases in which the distribution of $\bar{\boldsymbol{w}}_k$ is exactly Gaussian. For example, if there is only uncertainty in the measurement ($\Sigma_\omega = \Sigma_f = \Sigma_t = 0$), then that uncertainty propagates linearly through Equation 4.1 and preserves the Gaussian distribution. Alternatively, if ω_3 is zero and either Σ_ω or Σ_t is zero, then the resulting distribution is Gaussian. So, we can formulate the hypothesis test as

$$\begin{aligned} \mathcal{H}_0 &: \epsilon^2 < \gamma \\ \mathcal{H}_1 &: \epsilon^2 \geq \gamma \end{aligned} \tag{4.32}$$

where ϵ^2 is a metric based on the residual that we will define next and γ is the detection threshold. The term $\mathcal{H}_0$ denotes the null hypothesis that the object under consideration is a star while $\mathcal{H}_1$ is the hypothesis that the object is not a star. We will build our detection metric in a way that reduces the probability of a falsely classified star below a desired threshold. Consider the Mahalanobis distance of the residual from zero as the distance metric, $\boldsymbol{\epsilon}^2$ [85].

$$\epsilon^2 = \bar{\boldsymbol{w}}_k^T \Sigma_{\bar{w}_k}^{-1} \bar{\boldsymbol{w}}_k \tag{4.33}$$

The Mahalanobis distance squared has a chi-square distribution. The probability of false alarm, then, is the probability that the ϵ^2 metric will be greater than a particular value of γ. The probability that ϵ^2 exceeds γ is [88]

$$\mathrm{P}_{\mathrm{FA}} = \Pr\left[\epsilon_k^2 \geq \gamma\right] = e^{-\gamma/2} \tag{4.34}$$

So, we can rearrange the result above to solve for γ based on a desired probability of false alarm. The proper choice of γ is

$$\gamma = -2\ln\left(\mathrm{P}_{\mathrm{FA}}\right) \tag{4.35}$$

Figure 4.3 plots the proper choice of γ versus the desired probability of false alarm. Any object which was some apparent motion relative to the stellar background should return a gradually-increasing measurement residual. The precise rate of increase is dependent on the magnitude of the apparent motion. Slower objects such as near-Earth asteroids will require many more frames for a confident classification than a low-Earth orbiting object.

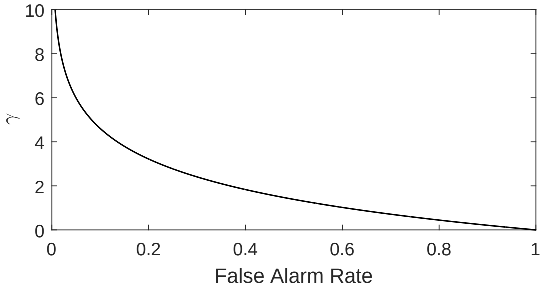

Figure 4.3: Choice of γ versus desired false alarm probability.

As with the formulation of the tracking problem, it is also possible to perform the classification process based on the full nonlinear differential equations of motion. The small field of view assumption introduces error which can lead to additional false positives depending on the magnitude of the neglected terms. Figure 4.4 shows a simulated scenario that violates the small field of view assumption. This scenario contains a single nonstar object in the center of the image. The classification process successfully detects the nonstar object, indicated by the red box. A number of objects around the edges of the image, however, are also falsely flagged as nonstars. This result is due to the neglected terms of Equation 3.55 that led to the linear solution in Equation 3.60. Objects further from the boresight move more quickly than expected and have higher measurement residuals as a result.

To solve the problem of false positive results due to the neglected wide field of view terms, we will also formulate a detector based on the full nonlinear equations of motion. The formulation will remain general enough to apply for any of the motion models in Chapter 3, but we will consider in particular the differential equation of motion describing an inertially-fixed object in view of a rotating sensor. See Equation 3.55 for the full model. There are two approaches we can take to build a continuous-discrete detector based on this model. The first uses the same linearization process as the Extended Kalman Filter (see Table 2.2.3). We are able to numerically propagate the state mean and propagate the covariance through linearized dynamics. Alternatively, we may use the Unscented Transform (see Table 2.2.3) to propagate the state mean and covariance. The remainder of the process of building the residual and detection metric is identical to that of the linear process above.

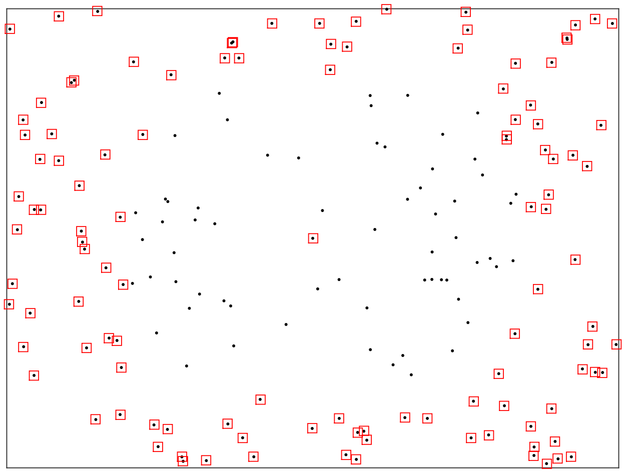

Figure 4.4: False positives in a simulated scenario due to the small FOV assumption.

First, let us derive the linearization-based continuous-discrete detector. Define the measurement residual in the following form:

$$\boldsymbol{w}_{k+1} = \boldsymbol{z}_{k+1} - \left(\boldsymbol{z}_k + \int_{t_{k-1}}^{t_{k+1}} f_c\left(\boldsymbol{\chi}_k\right) dt \right) \tag{4.36}$$

where $f_c\left(\boldsymbol{\chi}_k\right)$ is the right-hand side of Equation 3.55 (or another nonlinear solution from Chapter 3) and t_k is the time corresponding to the measurement $\boldsymbol{z}_k$. In words, we compute the predicted state from the previous measurement numerically and calculate its residual with the next measurement. This is effectively identical to our original definition, so the final steps of summing the residuals and computing the decision metric will be identical. The continuous-time relation for the covariance of a normally-distributed random variable, Σ_c, propagating through f_c is

$$\dot{\Sigma}_c\left(t\right) = F\left(\boldsymbol{\chi}\left(t\right), t\right) \Sigma_c\left(t\right) + \Sigma_c\left(t\right) F^T\left(\boldsymbol{\chi}\left(t\right), t\right) \tag{4.37}$$

with the initial covariance defined by the measurement and system uncertainties

$$\dot{\Sigma}_c\left(t_0\right) = \operatorname{diag}\left(\Sigma_z, \Sigma_\omega, \Sigma_f, \Sigma_t\right) \tag{4.38}$$

The term $F\left(\boldsymbol{\chi}\left(t\right), t\right)$ in Equation 4.37 describes the Jacobian of f_c with respect to the state variable $\boldsymbol{\chi}$.

$$F\left(\boldsymbol{\chi}\left(t\right), t\right) = \left.\frac{\partial f_c}{\partial \boldsymbol{\chi}}\right|_{\boldsymbol{\chi}(t)} \tag{4.39}$$

The Jacobian here is a 2 × 7 matrix in the way that we have formulated it. To input the Jacobian into Equation 4.37 it must be padded with zeros to convert it to a full 7 × 7. The equation for f_c must also contain equations describing the evolution of the other terms of the state variable $\boldsymbol{\chi}$. Since we assume that these terms are constant, then those terms in f_c

are all zero and thus the corresponding elements of the Jacobian are zero as well. So, after accumulating the measurement residuals according to Equation 4.36, the covariance of the resulting summation, $\boldsymbol{w}_{k+1}$, is

$$\Sigma_{w_{k+1}} = \Sigma_z + \Sigma_c \left(t_k + \Delta t_k\right) \tag{4.40}$$

The rest of the classification process is identical to the previous section. Though the linearization is only approximate, we will find that in simulated scenarios the additional fidelity from the higher-order motion models provides a benefit for wide fields of view.

The second method that we will derive relies on the Unscented Transform. As we discussed in Chapter 2, the Unscented Transform provides a method of approximating the covariance of a normally-distributed random variable when passed through a nonlinear function. In this case, the nonlinear function is Equation 3.55. The Unscented Transform works by choosing a specific set of sigma points to propagate forward numerically through the equation of motion. The resulting points allow for an estimate of the state mean and covariance. The usual choice of sigma points is [83]

$$\begin{aligned} \boldsymbol{\chi}_0^{[k]} &= \boldsymbol{\chi}\left(t_k\right) \\ \boldsymbol{\chi}_i^{[k]} &= \boldsymbol{\chi}\left(t_k\right) + \left(\sqrt{\left(n+k\right)\Sigma_{\boldsymbol{\chi}}}\right)_i \\ \boldsymbol{\chi}_{i+n}^{[k]} &= \boldsymbol{\chi}\left(t_k\right) - \left(\sqrt{\left(n+k\right)\Sigma_{\boldsymbol{\chi}}}\right)_i \end{aligned} \tag{4.41}$$

for $i = 1, \ldots, n$ where n is the dimension of the state variable. Note that $()_i$ signifies the ith column of the matrix contained in the parentheses. The term k is a tuning parameter that is typically set to $k = 3 - n$ [83]. For each time step, we propagate each sigma point forward to find a corresponding measurement prediction, $\boldsymbol{\mathcal{Z}}_i^{[k]}$, which is

$$\boldsymbol{\mathcal{Z}}_i^{[k]} = f_c\left(\boldsymbol{\chi}_i^{[k]}\right) \tag{4.42}$$

The same set of weightings that we saw in the Unscented Kalman Filter formulation in Table 2.2.3 allow for estimation of the state mean and covariance from these sigma points. The mean and covariance weightings, $W_i^{(m)}$ and $W_i^{(c)}$, are

$$W_i^{(m)} = \begin{cases} \lambda / \left(n+\lambda\right) & i = 0 \\ 1/\left\{2\left(n+\lambda\right)\right\} & i = 1, \ldots, 2n \end{cases} \tag{4.43}$$

and

$$W_i^{(c)} = \begin{cases} \lambda / \left(n+\lambda\right) + \left(1 - \alpha^2 + \beta\right) & i = 0 \\ 1/\left\{2\left(n+\lambda\right)\right\} & i = 1, \ldots, 2n \end{cases} \tag{4.44}$$

where $\lambda = \alpha^2\left(n+k\right) - n$, $\alpha \in [0,1]$, and β is a tuning parameter typically set to 2 [83]. The prediction mean resulting from the propagated set of sigma points and the weightings is

$$\bar{\boldsymbol{z}}_{k+1}^{-} = \sum_{i=0}^{2n} W_i^{(m)} \boldsymbol{\mathcal{Y}}_i^{[k]} \tag{4.45}$$

and the predicted covariance is

$$\Sigma_c^{[k]} = \sum_{i=0}^{2n} W_i^{(c)} \left\{ \boldsymbol{\mathcal{Z}}_i^{[k]} - \bar{\boldsymbol{z}}_{k+1}^{-} \right\} \left\{ \boldsymbol{\mathcal{Z}}_i^{[k]} - \bar{\boldsymbol{z}}_{k+1}^{-} \right\}^T \tag{4.46}$$

So, the measurement residual for this procedure is

$$\boldsymbol{w}_{k+1} = \boldsymbol{z}_{k+1} - \bar{\boldsymbol{z}}_{k+1}^{-} \tag{4.47}$$

and the residual's covariance is

$$\Sigma_{w_{k+1}} = \Sigma_z + \Sigma_c \tag{4.48}$$

The remainder of the classification process proceeds in the same way as in the natively linear process. This approach has the benefit over the linearized approach of not requiring computation of the Jacobian, which can be difficult to calculate analytically and computationally expensive to compute numerically [83].

Through simulation results we can illustrate the benefits of using the full nonlinear equations of motion over the approximate equations. The first test case that we consider here is a 1° field of view that satisfies the small field of view constraint. In the simulation, the field of view contains a single object. The angular rate of the sensor and time step have been chosen to cause a large displacement in the object – from the upper left corner to the bottom right – to incur as large an error as possible in the approximate linear solution from Equation 4.1 and highlight the differences between methods.

Table 4.1 contains the results for this simulation. Each of the three methods provided an estimate of the mean and covariance of the resulting distribution based on prescribed uncertainties in each of the system parameters. In addition, a 50,000 point Monte Carlo analysis provides the true distribution against which each of the methods are compared. To compare the predicted distributions and the true distributions, we use a Euclidean distance to compare the location of the mean and a Bhattacharyya distance to compare the uncertainties. The Bhattacharyya distance between two Gaussian distributions with mean μ_i and Σ_i for $i = 1, 2$ is [89]

$$D\left(\Sigma_1, \Sigma_2\right) = \frac{\left(\mu_1 - \mu_2\right)^T \Sigma^{-1} \left(\mu_1 - \mu_2\right)}{8} + \frac{1}{2} \ln \left(\frac{\det \Sigma}{\sqrt{\det \Sigma_1 \det \Sigma_2}} \right) \tag{4.49}$$

where

$$\Sigma = \frac{\Sigma_1 + \Sigma_2}{2} \tag{4.50}$$

In this first simulated case, all three methods exhibit comparable performance. The scenario preserves the small field of view assumption, therefore the approximate linear equation of motion provides a precise approximation of the true motion. The linearized method and the Unscented Transform do show slightly improved performance.

Table 4.1: Performance comparison for 1° FOV (normalized).

	Euclidean Distance	Bhattacharyya Distance
Linear	1.166	1.077
Linearized	1.001	1.038
Unscented	1	1

The second test case is set up in the same way, with an object moving from the top left corner of the image to the bottom right. In this scenario, however, the sensor has a 10° field of view which does not preserve the small field of view assumption. Table 4.2 contains the results from this simulation. Due to the increase in the field of view, the approximate linear equation of motion produces an estimate with a much higher error in both the mean and covariance of the distribution. The linearized method and the Unscented Transform, however, provide comparable performance to the 1° field of view scenario. Thus, as the field of view grows, so does the benefit from using the full equations of motion for object classification.

Table 4.2: Performance comparison for 10° FOV (normalized).

	Euclidean Distance	Bhattacharyya Distance
Linear	44.92	547.2
Linearized	1	1.036
Unscented	1.011	1

4.1.3 Simulated and Real-World Examples

Now we will look both a simulated and a real-world case for optical data analysis with the techniques discussed in the previous sections. This work has been implemented in a MATLAB tool intended to provide a general data reduction toolset. The code, for each input image, extracts all objects in view and passes that data into a multiple hypothesis Kalman filter bank following the theory described in the previous two sections. For more detail on the MATLAB implementation, see [68].

Figure 4.5 contains an annotated image from the simulated verification scenario. In this scenario, a sensor located in Blacksburg, Virginia observes an object in a geostationary orbit. The sensor is tasked to point at a location where the spacecraft is expected to be at some time in the future. The image in Figure 4.5 corresponds to the time when the spacecraft is at that point. The motion of Earth causes a slight precession in the background star field. The image is annotated to visualize the paths of each object contained in the image. Green dots indicate objects that have been classified as stars based the post-processing of the image data. The red box indicates the single space object, which has been successfully classified as a non-star object of interest by the algorithm. All of the yellow lines show the path of the corresponding object over the sequence of frames.

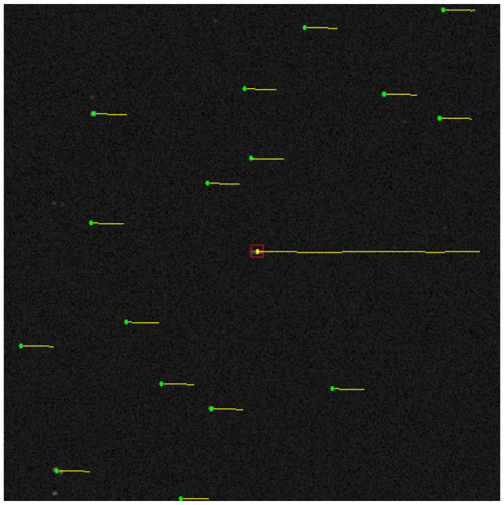

Figure 4.5: Annotated image of a simulated scenario.

Table 4.3 provides an overview of the optical parameters of the simulated sensor. The camera has a 1 degree field of view, satisfying the small field of view constraint of the motion models discussed earlier. The five second exposure time causes the object of interest to have some extent, but the background star field moves slowly enough that the stars remain point sources. The simulation covers a duration of 120 seconds.

Table 4.3: Simulated scenario camera parameters.

Image Resolution	512 × 512
Field of View	1°
Focal Length	500 mm
Integration Time	5 s

Figure 4.6 contains the Kalman filter residuals for the (x, y) position state of the non-star object of interest. Gray, dashed lines indicate the three sigma uncertainty bounds. The first detection of the spacecraft occurs 30 seconds into the simulation. In post-processing all tracks for the translation model in the filter bank initialize with an assumption of zero velocity and a large uncertainty. This leads to the transient phase seen in the first plot of Figure 4.6. The assumption of zero velocity leads to a large residual in the first step of 10 pixels. The next iteration reduces the residual to approximately 0.2 pixels and it remains within 0.6 pixels of zero for the remainder of the simulation. Note that, since the object's velocity is effectively all along the x-axis, the y coordinate residual does not contain a significant transient phase.

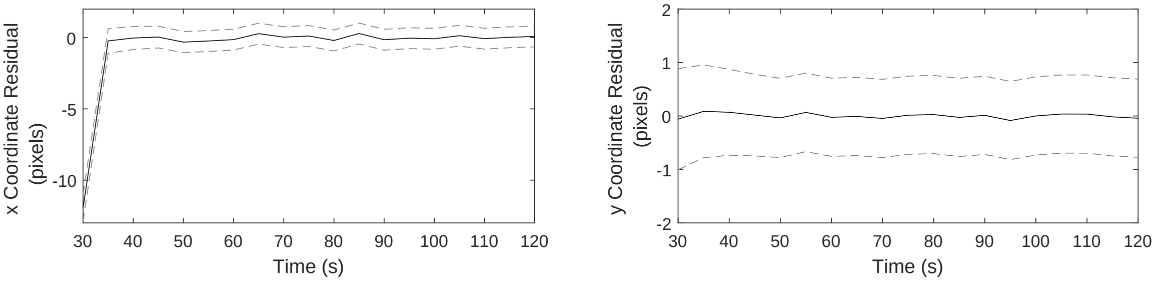

Figure 4.6: Kalman filter image plane position residuals for the simulated spacecraft.

At each time step, the algorithm reports a right ascension and declination corresponding to any object it detects in the frame. Since this is a simulated scenario, we can compare the reported values with the truth. Figure 4.7 contains the right ascension and declination residuals for the reported spacecraft observations. All of the measurements are within 2 arcseconds of the truth. The declination value is within about half of an arcsecond, while the right ascension exhibits a larger variance. The higher variance in the right ascension measurement is due to the fact that all of the objects motion in the image is along the x-axis. This axis is aligned with the direction of increasing right ascension, so the increased uncertainty from the motion of the object translates directly into increased uncertainty in that angle measurement.

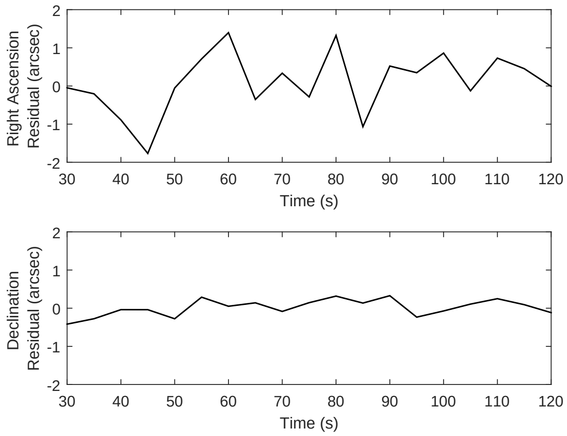

Figure 4.7: Measured RA/DEC versus truth residuals for the simulated scenario.

Figure 4.8 shows the x and y position residuals reported by the Kalman filter for a single star in the simulated case. Gray, dashed lines indicate the three sigma uncertainty bounds. The star motion model, as described in the previous sections, is initialized based on the current knowledge of the angular rate. This particular implementation begins with an initial guess for the angular rate and then updates its knowledge with a frame-to-frame rotation estimate and a median filter with a window size of 10 [90]. Due to error in the initial guess, the residuals for the star's Kalman filter also exhibits an initial transient phase. As the knowledge of the angular rate of the sensor approaches the true value, the Kalman filter residuals approach zero. The error in the Kalman filter prediction is proportional to the error in the angular rate estimate, so a bad estimate can lead to the loss of any track following the rotation-only motion model.

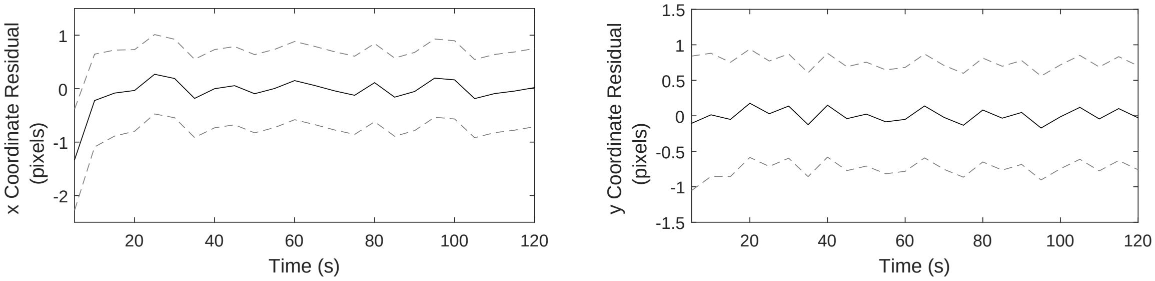

Figure 4.8: Kalman filter image plane position residuals for a simulated star.

In Figure 4.9, we see the result of the median filter on the magntidue of the angular rate at each point in time. The initial guess has a magnitude of approximately 5 arcseconds per second, while the truth is near 2 arcseconds per second. We see that the initial angular rate measurements cause a large jump in the overall estimate. This is due to the fact that the window of the median filter has not yet been populated. As more measurements come in the angular rate estimate converges toward the truth. With this particular implementation, error in the angular rate estimate sometimes causes the filter bank to lose track of stars. In extreme cases, the angular rate estimate will diverge, since the . A better approach is to perform a star identification whenever possible to infer the angular rate from the absolute attitude corresponding to each image.

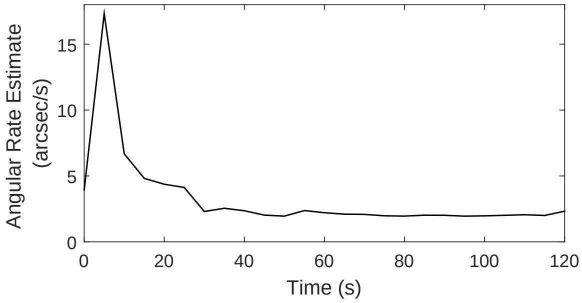

Figure 4.9: Online angular rate estimate for the simulated scenario.

Now let us consider this same process on a real-world data set. An annotated image from this data set is shown in Figure 4.10. This data is from a company called ExoAnalytic Solutions, made available in an online video format [9]. In the sequence of images, a sensor tracks the Mexsat-3 and Satmex-6 spacecraft [91]. During imaging, a piece of debris (a rocket body) passes through the field of view, crossing between the two objects in the image. So, the images contain two non-star objects which remain fixed in the image over the sequence as well as a third which enters the frame from the upper left-hand side.

This video contains a number of challenges beyond the simulated case. The images contain the noise from the original sensor in addition to compression artifacts, since the original raw data is not available. The sensor motion also includes some transient effects that violate the constant angular rate assumption. In fact, the latter portion of the original video has been removed due to a sudden delay jump in time. The background star field is relatively dense, so stars often corrupt the signal of the non-star objects.

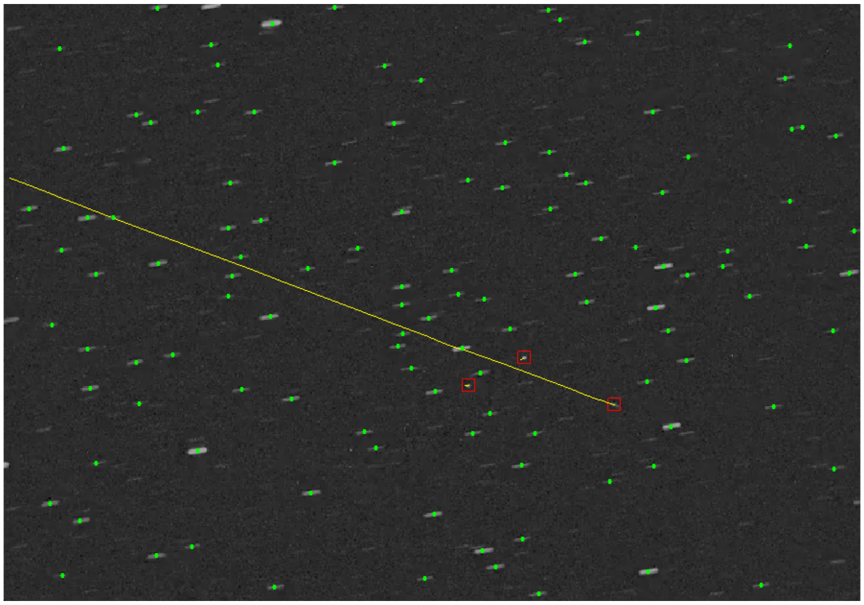

Figure 4.10: Annotated image from a real-world scenario.

Table 4.4 contains the camera parameters for the sensor as they are presented in the overlay of the original video. The field of view for this case is larger than in the simulated case, but we find that it still largely satisfies the small field of view assumption. Note that these camera parameters were not originally made available to the algorithm processing the data. In fact, precise knowledge of the camera parameters is not necessary; any arbitrary input is sufficient for tracking and classifying objects, however the output data (right ascension and declination) is only correct with the true values. Many of the required values may be estimated through a successful star identification [18].

As in the simulated case, we will examine the Kalman filter residuals for a number of objects. The first object we will consider is one of the static objects in the image frame. Figure 4.11 contains the x and y position prediction residuals for this object. With the initial guess of zero velocity for the translation model, there is no transient phase in either the x or y position residuals. There are some large jumps of the residual at a time about 1 minute

Table 4.4: Real-world scenario camera parameters [9].

Image Resolution	509 × 732
Field of View	2.32° × 1.75°
Focal Length	450 mm
Integration Time	10 s

after the start of the data set. These correspond to signal corruption due to overlapping star signals. In some cases, the overlapping star signals merge with that of the spacecraft. The detection is still made as if it were only the spacecraft, but the resulting centroid may be significantly off. The algorithm described here currently does not have a solution for differentiating overlapping signals.

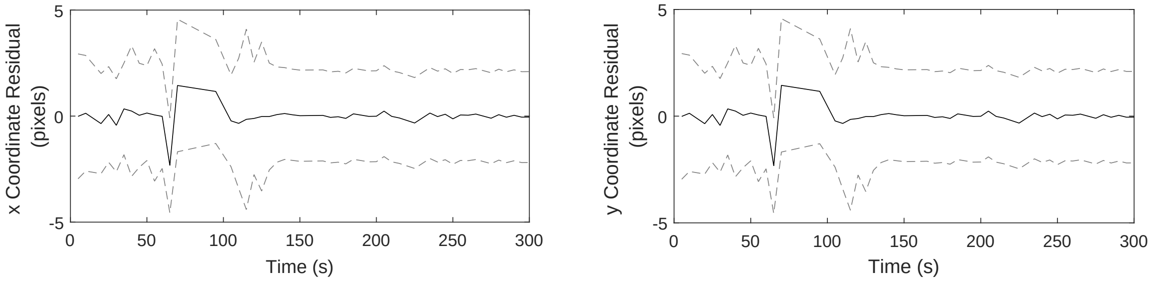

Figure 4.11: Image plane position residuals for a static RSO in real optical data.

The second set of residual plots, in Figure 4.12, corresponds to the object which enters the frame in the upper left-hand side of the image and moves between the two stationary spacecraft. This set of plots does show a transient phase in the first couple of detections where the algorithm is working to estimate the image-plane velocity of the object. After the second detection the prediction residual rapidly approaches zero. As in the case of the static non-star object, there are a number of points where this signal overlaps with a star signal, causing degradation in the resulting measurement. Overall, this signal has a much higher variance in the prediction residuals due to the additional localization uncertainty arising from its motion.

Finally, the third residual plot corresponds to a single star tracked throughout the entire sequence of frames. The initial guess for the angular rate is close to the true value, so there is no transient phase in the beginning of the scenario. The residual remains within 1 to 2 pixels of zero except for some poorly localized detections. The second spike in particular corresponds to an artifact in the motion of the sensor that causes a number of the Kalman filters tracking stars to momentarily lose track of the corresponding objects. For stars that were completely lost, the algorithm quickly initializes new filters and resumes tracking.

In summary, we see that the algorithm detailed in the previous two sections is capable of tracking stars and non-star space objects undergoing a variety of motion profiles simulta-

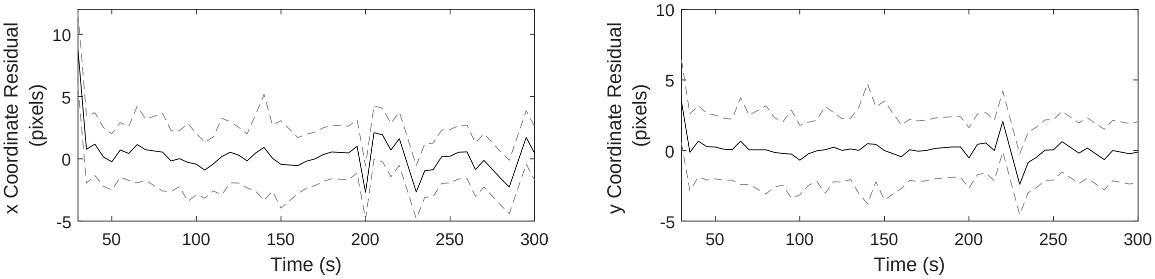

Figure 4.12: Image plane position residuals for a moving RSO in real optical data.

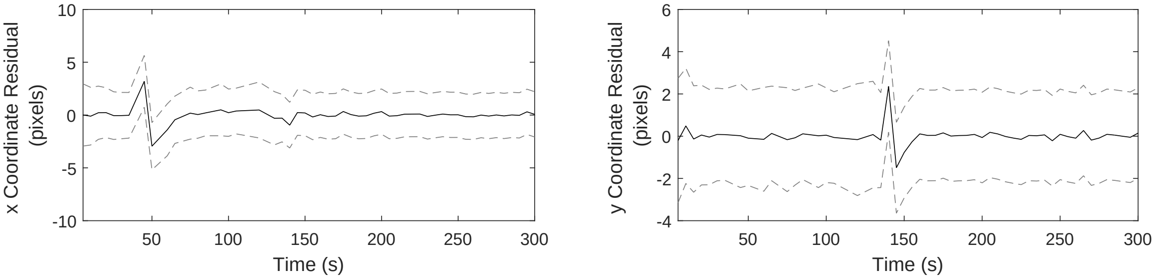

Figure 4.13: Image plane position residuals for a moving star in real optical data.

neously. This approach has been verified in both simulated and real data, however some challenges still exist. Specifically, methods for separating overlapping signals would help improve the overall operation of the filter bank.

4.2 Advanced Shift-and-Add

Another application for dynamic analysis of the image-plane motion of space objects is for shift-and-add procedures. The concept behind shift-and-add (also known as shift-and-stack or synthetic tracking) is that sequentially-captured frames are shifted and summed together in a way that simulates the effect of the sensor tracking an object of interest. This process is commonly used for detection of near-Earth asteroids (NEAs) [92, 31, 93], though it is applicable to detection of Earth-orbiting space objects as well [15]. These approaches commonly assume a constant velocity for objects of interest and shift subsequent images based on that known velocity. If the velocity is unknown, then it is possible to discretize a range of allowable velocity values and create image stacks based on all of those possible cases [15, 92, 31, 93]. This approach effectively searches a set of sequentially-captured images for all objects with constant velocity satisfying a range of possible parameters.

A key requirement of this technique is that the observing sensor has low read noise [39, 92, 31]. For example, the authors of [92] consider a sensor with a read noise of $1.2e^-$

(1.2 photoelectrons). In [7], the author estimates that a star with a visible magnitude of zero produces 19,100 photoelectrons per second per square millimeter of aperture area for a particular set of sensor parameters. Since the read noise is introduced to each image individually with a variance independent of the exposure time and signal brightness, the read noise effectively decreases the SNR of the signal for short exposures. If the read noise is high enough then the short exposures may actually reduce the SNR of the signal overall, effectively negating the benefit of the shift-and-add procedure.

To see why the shift-and-add process leads to a signal gain, consider a simplified model of an individual pixel in an optical sensor. Assume that this pixel is represented by normal distribution with some mean, $\bar{q}$, and standard deviation, σ_q.

$$q_k \sim N\left(\bar{q}, \sigma_q^2\right) \tag{4.51}$$

Assume also that the pixel contains a static signal that is not moving across the sensor. Though this is not always the case, the shift-and-add process works to force an object to remain static even when it has some apparent motion. Adding successive image frames leads to a pixel value of

$$\sum_{k=1}^{n} q_k \sim N\left(n\bar{q}, n\sigma_q^2\right) \tag{4.52}$$

where n is the number of frames. Note that this is also normally-distributed. The signal to noise ratio (SNR) of this pixel is [42]

$$\text{SNR} = \frac{n\bar{q}}{\sqrt{n\sigma_q^2}} = \sqrt{n}\frac{\bar{q}}{\sigma_q} \tag{4.53}$$

Thus the signal-to-noise ratio of the object of interest grows proportionally to the square root of the number of stacked images. Figure 4.14 shows the signal gain versus the number of stacked frames. Though this analysis is based on an idealized sensor model which is not completely representative of real sensors, it expresses the basic theory behind the shift-and-add process. The vertical dashed lines indicate the number of frames required for each whole-numbered SNR gain value from 1 to 7.

Now, we wish to develop an algorithm that leverages the dynamic analysis from Chapter 3 to perform a shift-and-add process in non-traditional cases where the shift cannot be represented by a constant velocity. We have seen from the solutions for a rotating sensor in Equation 3.22 and 3.25 that even a static object's velocity is not necessarily constant and is often dependent on its location in the image plane. Sensor drift around the boresight can cause a wide variation in the velocity of objects based on their location in the image plane. In general, we may write the shift in coordinates from one frame to the next as the $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{4.54}$$

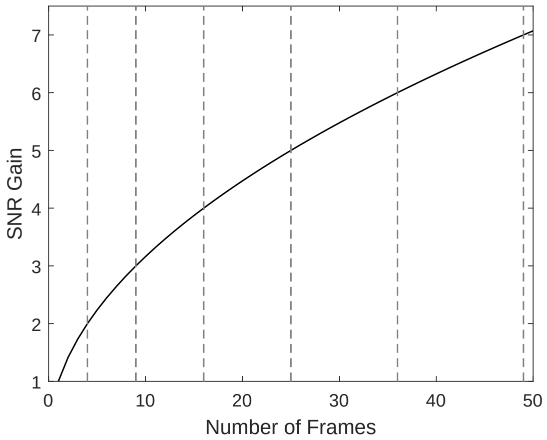

Figure 4.14: SNR gain versus number of stacked frames for a Gaussian distributed pixel.

where $i = 1, \ldots, n$ and n is the total number of pixels in the image. In words, this means that an object located at the coordinate (x, y) in the initial image moves to (x', y') in the subsequent image. So, if we represent the signal on the focal plane as a continuous function of x, y, and time, we can say that

$$I(x, y, t) = I(x', y', t + \Delta t) \tag{4.55}$$

where Δt is the time step between frames. Note that I is a continuous representation of the image and not the true image. The image data is actually a discrete sampling of I. This equation gives the relationship between one image and the next and allows us to manipulate subsequent frames in a way that removes the effect of any undesired motion. The equation representing the desired stack of images is

$$I_{\text{stack}}(x, y) = I(x, y, t_0) + \sum_{k=1}^{\ell} I(x', y', t_0 + k\Delta t) \tag{4.56}$$

where ℓ is the number of shifted images. If $x = x'$ and $y = y'$ (i.e. there is no undesired motion), then the equation above simplifies to a direct summation of images. Now, let us proceed with an example to demonstrate this advanced shift-and-add process. We will consider a simulated scenario with a sensor defined by the parameters in Table 4.5.

Table 4.5: Simulated camera parameters for the advanced shift-and-add scenario.

Image Resolution	512 × 512
Field of View	1° × 1°
Focal Length	500 mm
Integration Time	2 s

Figure 4.15 contains two images of the star field from this simulation. The first image contains no noise to highlight the stars that are in view. The stars for this simulation are taken from the Tycho-2 catalog [6]. The second image in Figure 4.15 is the same star field with Gaussian random noise added to each pixel. In this image most of the stars have fallen below the noise floor and only a few bright stars remain.

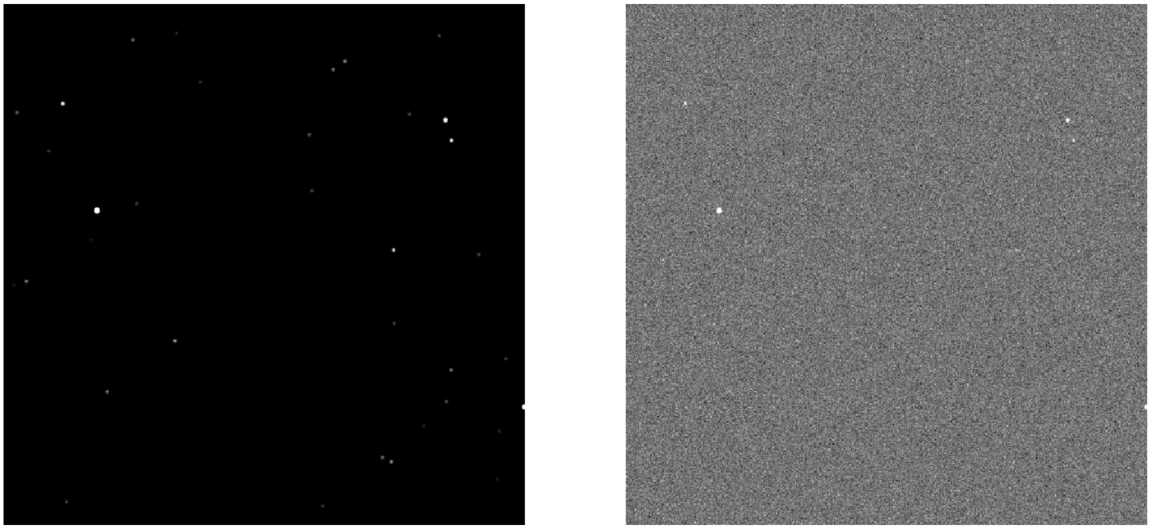

Figure 4.15: Simulated star field (left) and the same field with Gaussian noise (right).

For this verification scenario we will assume that the goal of the shift-and-add process is to extract the dim star signals. Star signals are commonly required for precise pointing estimates of an optical sensor [17]. During imaging the sensor is rotating about a constant rotation axis with an angular rate of

$$\boldsymbol{\omega} = [0, 0, 1]^T \ \text{deg/s} \tag{4.57}$$

In words, the rotation is entirely around the boresight at a rate of 1 degree per second. This motion profile might occur on a spin-stabilized spacecraft or on an on-orbit optical sensor which has lost some control authority. A well-known example of such a spacecraft is the Kepler telescope, which lost authority over two reaction wheels and 2012 and 2013 [94]. Figure 4.16 contains a direct summation of the images produced by the simulated sensor. The sensor captures images with a 3 second interval and a two second exposure time. In all, the sensor captures 11 images and rotates approximately 30 degrees over the simulation time. Notice that, since there is no overlap in the stars in Figure 4.16 from one frame to the next, there is no signal gain from the direct summation of frames.

Recall the Equations of motion describing the motion of an intertially-fixed object across the image plane of a sensor rotating with a constant angular velocity. From Equation 3.22

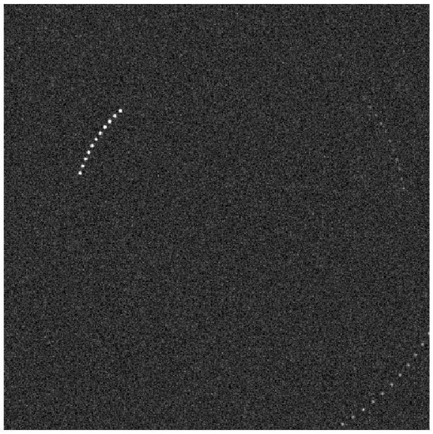

Figure 4.16: Direct sum of the simulated 11-image sequence.

and 3.25 we have

$$
\begin{aligned}
\dot{x} &= \omega_3 y + f\omega_2 + \frac{1}{f}\omega_2 x^2 - \frac{1}{f}\omega_1 xy \\
\dot{y} &= -\omega_3 x - f\omega_1 + \frac{1}{f}\omega_2 xy - \frac{1}{f}\omega_1 y^2
\end{aligned}
\tag{4.58}
$$

In an effort to include a full representation of the motion of the sensor, we will use these equations in full rather than the approximate solutions from Equation 3.60. The equations above are predicted numerically for the purposes of this simulation. So, x' and y' are

$$
\begin{aligned}
x'(t) &= \int_{t_0}^{t} \dot{x}dt + x(t_0) \\
y'(t) &= \int_{t_0}^{t} \dot{y}dt + y(t_0)
\end{aligned}
\tag{4.59}
$$

where $t_0 = 0$ is the start time of the simulation. For each image, we predict the focal plane coordinate of each pixel forward from $t_0 = 0$ to the current simulation time. Equation 4.56 describes how the each image is stacked based on the predicted coordinates (x', y'). Since Equation 4.56 treats the data as a continuum, when in fact the image data is discrete, a linear interpolation provides intensity values for coordinates between data points. Figure 4.17 contains the resulting image stack. In the figure, the first image is the true, noise-free star field. The second image is the result of the advanced shift-and-add process. It is immediately apparent that a number of stars that were not visible in the noisy image in Figure 4.15 have moved above the noise floor while the stars that were already visible became brighter. The brightest star – toward the upper left of the image – saw a peak signal gain of approximately 5.6 times its original intensity.

It is important to note that this technique is extremely computationally intensive. In the most basic formulation, each pixel must be propagated from the reference time step ($t_0 = 0$

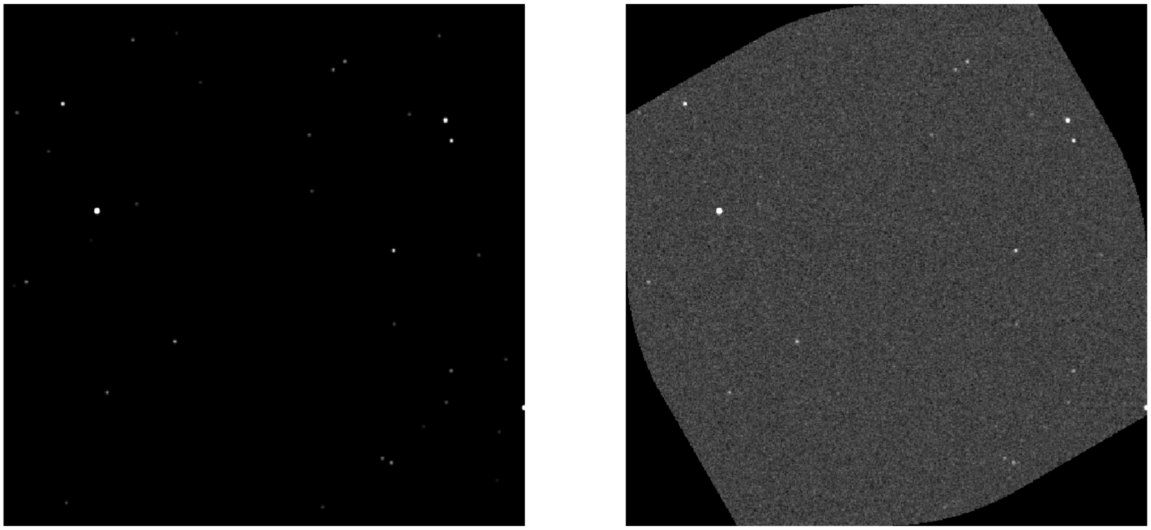

Figure 4.17: Result of 11 stacked frames (right) beside the true star field (left).

in this case) to the current time. For the simulated example there are $262,144$ individual pixels. Conveniently, this process is easily parallelizable. Propagation of a single focal plane coordinate is only dependent on the initial position, the angular rate, and the focal length of the sensor, so all of them can theoretically be propagated simultaneously. Some approximations may also be beneficial to this process. First, the approximate solution in Equation 3.60 removes the need for a costly numerical integration. Further, propagation of every pixel location in the image is not necessarily required. One might choose instead to predict every other pixel forward and approximate the intermediate values with an interpolation.

Another characteristic of this process that might be leveraged is the fact that, for a constant angular rate and time between frames, the frame-to-frame mapping is constant. Since the algorithm predicts forward the locations of the pixels on the image plane at each time step, the starting location is the same for each frame. It follows then that the prediction will also be the same for an identical angular rate and step time. In that case, we only need to compute the mapping between frames once. Subsequent iterations can then use the stored mapping with a greatly reduced computational cost.

Also of note is the fact that interpolation of the raw image data introduces additional error due to the presence of noise in the image and the innate inaccuracy of the interpolation process. In the raw image data, each pixel contains a certain amount of noise from various sources. After interpolation, the noise in a single pixel incorporates the noise of at least 4 neighboring pixels, possibly many more depending on the interpolation method. This effect likely reduces the signal gain from the shift-and-add process, but in the simulated cases here the stack of frames does still show an appreciable gain.

The concept of removing the rotation of the sensor from frame to frame leads to another interesting effect when there is a non-star object in the frame. Figure 4.18 contains the same simulated scenario as before, except with a non-star object exhibiting some apparent motion of its own. The first image of the figure shows a direct summation of the sequence of images while the image on the right contains the result of the advanced shift-and-add procedure. The shift-and-add process uses the same equations of motion as before (Equation 4.58), so the motion of the non-star object is not negated. The shift-and-add process in this case serves only to remove the rotation of the sensor. In the resulting image stack, the stars receive the same signal gain as in the previous case. The non-star object, due to its own apparent motion, is not stacked perfectly. So, the non-star object retains only its apparent motion while the rotation of the sensor has been negated.

This correction, then, effectively removes the motion of the sensor. Further, this implies that the objects position in the image plane is fully defined by their own motion in the inertial reference frame. Recall the equation of motion for such objects from Chapter 3: Equation 3.106. Thus, the advanced shift-and-add procedure also provides a way to augment the filtering techniques discussed in the previous section. If we transform each frame in a way that removes the rotation of the sensor, then we are able to use the motion models for a non-rotating sensor as the basis of a Kalman filter bank.

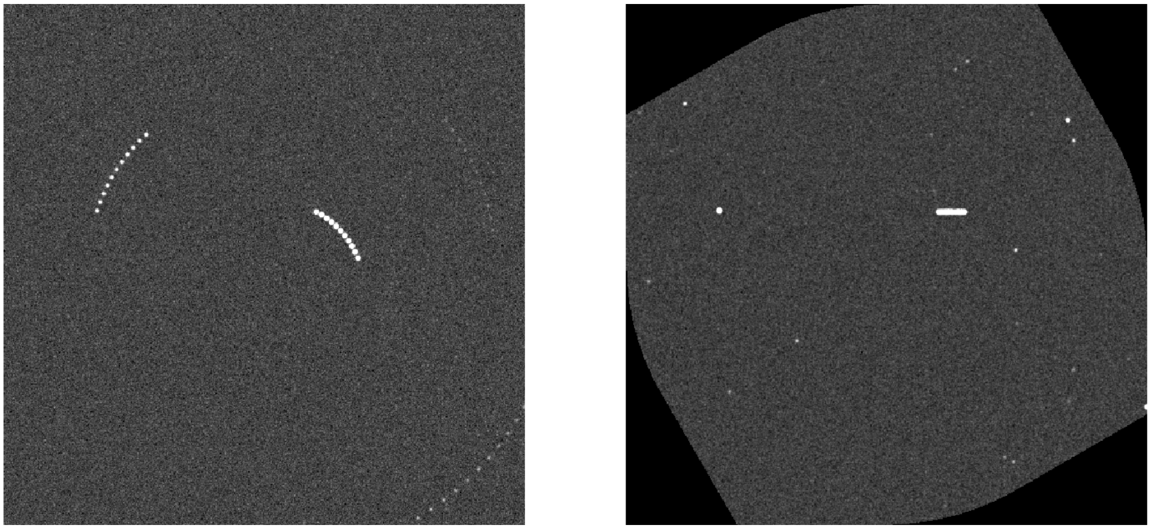

Figure 4.18: Result of 11 stacked frames (right) beside directly summed frames (left).

4.3 Template-Based Star Subtraction

The previous section detailed an advanced shift-and-add process based on the focal plane motion models derived in Chapter 3. A drawback of this approach to image analysis is the presence of star clutter. When an object of interest is not a star and exhibits some apparent motion of its own it becomes possible to observe overlapping signals. Though the shift-and-add process will increase the SNR of the object of interest, conflicting signals may still make it impossible to extract.

Figure 4.19 contains a simulated scenario that illustrates this problem. This simulated sequence of images contains a single non-star object that is fixed in the center of the image. In order to keep this space object in the center of the image, the sensor is slewing with a prescribed angular rate, causing the stellar background to move across the image. In addition, the space object is below the detection threshold in a single image, so the observer must capture several images sequentially. In the final summation of the images the object is above the noise floor. The star streaks clutter the resulting image stack, however, including two streaks which are directly occluding the object of interest. The work in this section addresses the problem of removing the undesired signals to leave only the object of interest. This work will leverage the dynamic motion models from Chapter 3 to predict the appearance of the star signals. Chapter 5 will take this analysis a step further and leverage geometric image analysis to enable this method even when the star signals have significant curvature.

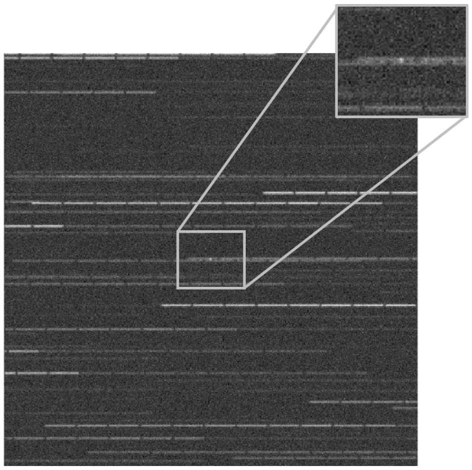

Figure 4.19: Space object (center) with star clutter from 20 sequentially-stacked images.

There are a number of methods in the literature that can allow for the removal of star signals. The first, and most basic, approach involves direct comparison of successive frames and can operate in one of two ways. The first assumes that the stars are static and any object of interest is moving across the frame. If this assumption holds true, then subtraction of sequential frames should decrease the SNR of the star signals while preserving any object which did not remain fixed [27]. Alternatively, analysis of temporal variations in individual pixels can give some insight into whether that pixel contains a star. The Wide Area Space Space Surveillance System (WASSS) analyses the variance of an individual pixel's intensity over time. In the WASSS implementation, stars move with the sidereal rotation rate of Earth and the objects of interest – geosynchronous spacecraft – are fixed. Any pixel which contains a transient signal, then, corresponds to a star [30]

Another technique for handling overlapping signals is a process known as deblending. Deblending routines aim to separate overlapping signals through analysis of the combined data. There are a number of different methods for deblending which [95] discusses in detail. The most commonly applied deblending method is included in the popular source extractor software package, which is a general software tool for reducing unresolved space imagery [40]. The Source Extractor software package analyzes combined signals through an iterative thresholding process. At each threshold level, the algorithm determine the number of possible objects that are present. Analysis of the resulting tree structure determines which components of the combined signal belong to a particular object. This process is often effective at separating objects, however [96] describes deblending as "dangerous" because the threshold process has a tendency to break elongated objects into multiple pieces even when there are no overlapping signals.

Perhaps the most promising technique for star removal is a point-spread function (PSF) fitting process [96]. The PSF fitting approach aims to approximate the a single signal in an image as a linearly-extended point source. A nonlinear fitting process varies the characteristics of the signal fit to best match the data at a particular location. This process is commonly used as a precise way to localized objects in an image, with a typical error less than a third of a pixel and sometimes as low as 0.01 pixels [96]. This approach is closely related to the method we will derive in this section.

4.3.1 Star Subtraction as an Optimization Problem

The PSF fitting method in [96] formulates the process of iteratively fitting an unknown signal through a nonlinear process. Though this process is not typically used for star removal, if the result is truely representative of the underlying data then it should be sufficient to remove the signal by simply subtracting the result. So, we will consider this technique as a star removal process. The algorithm assumes that objects in the image have a Gaussian point spread function with a signal amplitude and width that are variable. Assume that the point spread function, which describes the appearance of a static object point source in an image for a particular sensor, is $f_{\text{point}}\left(\cdot\right)$ where

$$f_{\text{point}}\left(r, A, w\right) = A \exp\left(-4\frac{r^2}{w^2}\ln 2\right) \tag{4.60}$$

In the equation above, A represents the amplitude (brightness) of the signal, w is the full width at half maximum (FWHM), and r is the radius from the center of a point of interest. The Full width at half maximum is the distance from the peak signal intensity at which the signal drops to half of the peak value [96]. Now, $f_{\text{point}}\left(\cdot\right)$ only describes the appearance of a signal in the image plane at a particular instant in time. Since optical sensors work by collecting photons over time, any apparent motion in that signal will cause it to deviate from $f_{\text{point}}\left(\cdot\right)$. If we assume that a signal is moving with constant velocity in the image plane, then the resulting signal from a long exposure will be $f_{\text{point}}\left(\cdot\right)$, but extended linearly along the direction of motion. We may write this as

$$f_{\text{trail}} = f_{\text{point}}\left(\sqrt{\left[s\left(|x| - l/2\right)\right]^2 + y^2}, A, w\right) \tag{4.61}$$

where l is the length of the signal and $s(x)$ is

$$s(x) = \begin{cases} 0, & \text{if } \; x \leq 0 \\ x, & \text{if } \; x > 0 \end{cases} \tag{4.62}$$

This relation effectively stretches the PSF along the x direction. For generality, we may also consider the same PSF rotated by some angle θ. The rotated PSF is given by [96]

$$f(x, y, A, w, l, \theta, x_0, y_0) = f_{\text{trail}}(x', y', A, w, l) \tag{4.63}$$

where

$$\begin{aligned} x' &= (x - x_0)\cos\theta + (y - y_0)\sin\theta \\ y' &= (y - y_0)\cos\theta + (x - x_0)\sin\theta \end{aligned} \tag{4.64}$$

With this definition of the PSF, the optimal signal fitting problem is

$$\text{minimize} \qquad \textstyle\sum_x \sum_y \left[I(x, y) - f(x, y, A, w, l, \theta, x_0, y_0)\right] \tag{4.65}$$

In words, we wish to minimize the difference between the image data, $I(x, y)$, and the assumed PSF, $f(x, y, A, w, l, \theta, x_0, y_0)$. The placement of the PSF, (x_0, y_0), the angle, θ, the length, l, the amplitude, A, and the FWHM, w, are all parameters which we are able to vary to improve the fit. In [96], the author uses a Levenburg-Marquardt algorithm to solve this problem for a single signal at a time. Due to the inherent nonlinearity of the problem and the iteration required to solve it, solution can be expensive, especially for a large number of signals. Further, this PSF fitting approach may fail for overlapping signals [96].

The method detailed in the following sections provides a more efficient, global approach at the expense of some of the fidelity of the technique above. We have already seen from Equation 3.62 that in the absence of drift around the boresight of the sensor, all star signals in the image have approximately the same length. Incorporating the assumption from above that the point source is consistent across the image, the solution implies that all signals in the image are identical. Thus, it is possible to build a single representative template that describes the appearance of a star signal anywhere in the image. The method described in the following sections will leverage this fact to produce a global, convex formulation for the star subtraction problem.

4.3.2 A Class of Convex Optimization Problems

In this section we will consider a class of general formulations for a global star removal process. In particular, we will consider formulations which lead to a convex optimization problem. A convex formulation is beneficial because it guarantees a global minimum and convergence in a finite number of iterations [97]. Such a formulation will greatly reduce the computational cost of the star subtraction routine from the local, nonlinear method in [96].

Before we begin let us define some of the variables and conventions that we will use to formulate this problem. First, consider the image data. Up until now we have typically treated images as continuous data defined as a function of the x and y image plane positions. In this section, we will handle the image data in its true discrete format. Figure 4.20 illustrates the convention for representing images in the problem formulations that follow. In this representation, images are treated as vectors, rather than matrices. This differs from the way that we have considered images in some of the past sections; here we handle them in the true discrete form that they are collected. Expressing the image as a vector rather than a matrix is beneficial mathematically, as we will see. The images are "unrolled" in a way that stacks each column of the raw data on the column next to it.

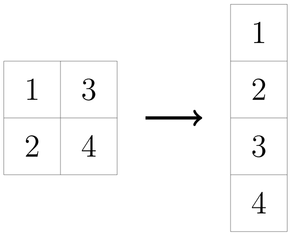

Figure 4.20: "Unrolling" an image to create a vector representation.

Here we will use three different types of image vectors. The first is the raw image data, denoted by $I \in \mathbb{R}^{mn}$ for an $m \times n$ image. The second is a background fit, denoted by $H \in \mathbb{R}^{mn}$. The background fit term is not always necessary, but adds some generality to the problem. There are a number of background fitting procedures which aim to characterize structure in the background signal [51, 52]. These techniques aim to subtract the effect of non-homogeneous backgrounds to simplify processing. Since a background subtraction process directly modifies the image data, it will be closely coupled to the star removal problem. A star template image, $T \in \mathbb{R}^{mn}$, will be the third type of image vector that we will consider. Though a single star signal will likely only cover a small portion of the total image, templates for each potential star in the image will be constructed according to the dimensions of the overall image. We will see later why this approach is beneficial.

Finally, the term $\boldsymbol{\nu} \in [\ell \times 1]$ will define a template amplitude scaling for the fitting process. Here, ℓ is the total number of star templates. Figure 4.21 contains an illustration of the function of the term $\boldsymbol{\nu}$ in the fitting process. In this Figure the shaded gray portion indicates a cross section of the intensity data for a single streaked star signal. The dotted line indicates an idealized template. The term $\boldsymbol{\nu}$ scales the idealized template to best fit the underlying data. This is the basic principle behind the star fitting approach described in this section.

With these definitions in mind, let us consider the most basic formulation based only on

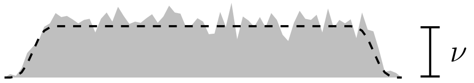

Figure 4.21: A sample cross-section of a noisy star signal (gray) with an idealized signal fit.

the physics of optical sensors. We wish to approximate the underlying star signals in an image, which means we would like to minimize the difference between the image data and a set of scaled template images. Since optical sensor provide a measure of the photons that contacted the focal plane during a given exposure time, the signal in any pixel cannot be negative [39]. So a basic formulation is

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T \left(I - H - A\boldsymbol{\nu} \right) \\ \text{subject to} & I - H - A\boldsymbol{\nu} \succeq 0 \\ & \boldsymbol{\nu} \succeq 0 \end{array} \tag{4.66}$$

where $A = [T_1, \ldots, T_\ell]$ is a matrix where each column contains a single template image and $\mathbf{1}^T$ is an $m \times n$ vector of ones. This formulation attempts to minimize the summed difference between each pixel in the image data and the template data. Since I and H are constant, an equivalent problem is

$$\begin{array}{ll} \text{minimize} & -\mathbf{1}^T \left(A\boldsymbol{\nu} \right) \\ \text{subject to} & I - H - A\boldsymbol{\nu} \succeq 0 \\ & \boldsymbol{\nu} \succeq 0 \end{array} \tag{4.67}$$

In words, we can equivalently attempt to maximize the the sum of the pixels in the scaled template images subject to the constraint that the difference between the image, background fit, and template images cannot become negative. Note that this problem is only feasible for $I \preceq H$ otherwise the first constraint is violated for any choice of $\boldsymbol{\nu}$. This optimization is linear and convex so it is guaranteed to have a global minimum and converge in a finite number of iterations [97]. Another approach to formulating the star removal problem is to include a normed objective function. The following optimization problem attempts to minimize the normed difference between the image, background fit, and template images.

$$\begin{array}{ll} \text{minimize} & \left\| I - H - A\boldsymbol{\nu} \right\|_p \\ \text{subject to} & \boldsymbol{\nu} \succeq 0 \end{array} \tag{4.68}$$

We still include the constraint that the scaling vector must be positive to produce a result that is guaranteed to make physical sense. For generality, we have used the p-norm. We will consider two norms in particular, the 1-norm and the 2-norm. First, let us slightly modify the problem above to put it in a simpler form. If we relax the constraint that $\boldsymbol{\nu} \succeq 0$, then the problem becomes

$$\text{minimize} \quad \left\| I - H - A\boldsymbol{\nu} \right\|_p \tag{4.69}$$

We still desire a scaling vector, $\boldsymbol{\nu}$, that is positive, but relaxation of the constraint will lead to an optimization problem that is much easier to solve. In practice, we will find that careful placement of the templates will avoid any problems that arise from the relaxed constraint. If a template is misplaced, however, the subtraction may introduce additional artifacts into the result. Figure 4.22 contains a simulated exampled of one such case. In this simulation, a template for a streaked star was placed on a non-star point source signal. The template does not match the underlying signal, but instead of leaving the scaling at zero, the optimization process chose a negative scaling term and removed a portion of the noise. This case used the 2-norm object function. The 1-norm objective function reduces the effect of misplaced templates because minimization of the 1-norm tends to produce sparse solutions [97].

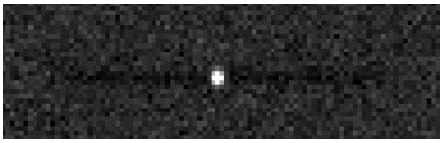

Figure 4.22: Artifacts incurred from misplaced template and relaxed constraint.

Now, let us look at some alternative constructions for specific norms in the objective function. First, the 1-norm optimization problem is

$$\text{minimize} \qquad \|I - H - A\boldsymbol{\nu}\|_1 \tag{4.70}$$

We may also construct this as a linear problem. If we define the slack variable, $\boldsymbol{t}$, we can write [97]

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T\boldsymbol{t} \\ \text{subject to} & I - H - A\boldsymbol{\nu} \succeq -\boldsymbol{t} \\ & I - H - A\boldsymbol{\nu} \preceq \boldsymbol{t} \end{array} \tag{4.71}$$

This problem attempts to minimize the sum of the elements of the slack variable $\boldsymbol{t}$ subject to constraints on each element of $\boldsymbol{t}$. The constraints force the slack variable to be greater than or equal to the absolute value of each element of the interior portion of the 1-norm in Equation 4.70. So, the minimum of the problem above will enforce that each element of $\boldsymbol{t}$ is equivalent to the absolute value of each element of $(I - H - A\boldsymbol{\nu})$. Further, the sum of the elements of $\boldsymbol{t}$ will be equivalent to the 1-norm of $(I - H - A\boldsymbol{\nu})$. So, this formulation is equivalent to that in Equation 4.70, but is more efficient to solve [97].

The 2-norm template-fitting problem ends up being the most straight-forward to solve, but it also has the most potential for artifacts like the one in Figure 4.22. The 2-norm problem formulation is

$$\text{minimize} \qquad \|I - H - A\boldsymbol{\nu}\|_2 \tag{4.72}$$

which is a least-squares optimization problem. Conveniently, a closed-form solution to this problem is well-known. The optimal choice of $\boldsymbol{\nu}$ is [97]

$$\boldsymbol{\nu}^* = \left(A^T A\right)^{-1} A^T (I - H) \tag{4.73}$$

4.3.3 Building Star Templates

Equally important to the formulation of the optimization problem is the construction of the templates used for the star removal process. Recall the approximate equation describing the motion of an inertially-fixed object across the image plane of a rotating sensor. Equation 3.62 shows that, when there is no rotation about the boresight (and therefore no curvature in the star signals) a star's motion evolves according to

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x(t_0) \\ y(t_0) \end{bmatrix} + f \begin{bmatrix} \omega_2 \\ -\omega_1 \end{bmatrix} t \tag{4.74}$$

In Chapter 5, we will see that this process is also possible when $\omega_3 \neq 0$. The equation above is represented in the focal plane coordinate system as a physical distance from the origin in the center of the CCD (See Figure 2.5). To build the templates, we will convert these equations of motion to the pixel coordinate system through Equation 2.5. The new equations describing the motion of a star are

$$\begin{bmatrix} \bar{x}(t) \\ \bar{y}(t) \end{bmatrix} = \begin{bmatrix} \bar{x}(t_0) \\ \bar{y}(t_0) \end{bmatrix} + \frac{f}{s} \begin{bmatrix} \omega_2 \\ -\omega_1 \end{bmatrix} t \tag{4.75}$$

Notice that the equation above implies that all star signals have an identical appearance in the image, regardless of their location on the focal plane. We will leverage this fact to simplify analysis; only a single template is necessary to fit all of the stars in an image. Now, with a description of the motion of a star in mind, we must represent the appearance of a star in the resulting image. Consider a general point spread function, $f_{psf}(\eta, \xi)$. This function describes how the appearance of a point source signal varies further from the center of the true location of the signal at $(0, 0)$. The rate of change of intensity of the pixel located at $(\bar{x}, \bar{y})$ due to a star with position $(\bar{x}_0(t), \bar{y}_0(t))$ at time t is

$$\dot{T}(\bar{x}, \bar{y}, \bar{x}_0(t), \bar{y}_0(t)) = \int_{\bar{y}-0.5}^{\bar{y}+0.5} \int_{\bar{x}-0.5}^{\bar{x}+0.5} f_{psf}(\bar{x} - \bar{x}_0(t), \bar{y} - \bar{y}_0(t))\, d\bar{x} d\bar{y} \tag{4.76}$$

Note that, for now, we return to the convention that images are treated as a continuum, but only evaluated at discrete pixel locations. So the construction of the template will be derived as if it were a continuum and then handled discretely in practice. Now, let us consider an example PSF. If we assume that a point source follows a symmetric Gaussian distribution, the PSF function is

$$f_{psf}(\bar{x}, \bar{y}, \bar{x}_0(t), \bar{y}_0(t)) = \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{[\bar{x} - \bar{x}_0(t)]^2 + [\bar{y} - \bar{y}_0(t)]^2}{2\sigma^2} \right\} \tag{4.77}$$

Notice that there is no term to include an amplitude of the signal in f_{psf}. The amplitude is unnecessary because the template will be scaled during the optimization process to approximate the true intensity of the data. Later on we will normalize the template to ensure a maximum value of 1. With this choice of f_{psf}, the template's rate of change, $\dot{T}$, is

$$\dot{T}\left(\bar{x},\bar{y},\bar{x}_0\left(t\right),\bar{y}_0\left(t\right)\right)=\frac{1}{2\pi\sigma^2}\int_{\bar{y}-0.5}^{\bar{y}+0.5}\int_{\bar{x}-0.5}^{\bar{x}+0.5}\exp\left\{-\frac{\left[\bar{x}-\bar{x}_0\left(t\right)\right]^2+\left[\bar{y}-\bar{y}_0\left(t\right)\right]^2}{2\sigma^2}\right\}d\bar{x}d\bar{y} \tag{4.78}$$

Though there is not a closed-form solution to the integral of a Gaussian distribution, we can write the equation above in the form

$$\begin{aligned}\dot{T}\left(\bar{x},\bar{y},\bar{x}_0\left(t\right),\bar{y}_0\left(t\right)\right)=\frac{1}{4}&\left[\operatorname{erf}\left(\frac{\bar{x}-\bar{x}_0-0.5}{\sqrt{2}\sigma}\right)+\operatorname{erf}\left(\frac{\bar{x}_0-\bar{x}-0.5}{\sqrt{2}\sigma}\right)\right]\\&\cdot\left[\operatorname{erf}\left(\frac{\bar{y}-\bar{y}_0-0.5}{\sqrt{2}\sigma}\right)+\operatorname{erf}\left(\frac{\bar{y}_0-\bar{y}-0.5}{\sqrt{2}\sigma}\right)\right]\end{aligned} \tag{4.79}$$

where $\operatorname{erf}\left(\cdot\right)$ denotes the error function. The error function cannot be represented by elementary functions, but lookup tables for $\operatorname{erf}\left(\cdot\right)$ are widely available and allow for efficient computation. Thus we are able to quickly determine the instantaneous contribution of a star signal to a particular pixel. Now, since optical sensors collect photons over time, we must integrate the rate of change of each pixel to obtain the final signal intensity. So, the template intensity as a function of only the pixel locations $\left(\bar{x},\bar{y}\right)$ is

$$T\left(\bar{x},\bar{y}\right)=\int_0^{\Delta t}\dot{T}\left(\bar{x},\bar{y},\bar{x}_0\left(\tau\right),\bar{y}_0\left(\tau\right)\right)d\tau \tag{4.80}$$

This continuous function describes the intensity of a theoretical pixel centered on the coordinate $\left(\bar{x},\bar{y}\right)$ due to an object moving over the sensor with position $\left(\bar{x}_0\left(t\right),\bar{y}_0\left(t\right)\right)$. To convert to a discrete representation we simply evaluate $T\left(\bar{x},\bar{y}\right)$ at only discrete pixel locations within the bounds of the image plane. Finally, we normalize the resulting template to a maximum value of one.

$$\bar{T}\left(\bar{x},\bar{y}\right)=\frac{T\left(\bar{x},\bar{y}\right)}{\max_{\bar{x},\bar{y}}\left\{T\left(\bar{x},\bar{y}\right)\right\}}\in\left[0,1\right] \tag{4.81}$$

With all of the templates normalized to one, the scaling parameter, $\boldsymbol{\nu}$ will directly correspond to the amplitude of the underlying signal.

4.3.4 Simulated Results

Now let us consider a simulated scenario to observer the results of this process on realistic data. Table 4.6 describes the optical sensor in the simulated scenario. In this simulated case, the sensor captures images with a 3 second exposure time and a 0.05 second day to emulate processing time within the sensor. Throughout the scenario, the sensor tracks a space object based on prior knowledge. The space object is too dim to be detected in a single frame, however, so the frames are stacked to bring the signal above the noise floor. The result of this process was seen in Figure 4.19, at the beginning of this section. After stacking 20 images, the object of interest does show a signal gain, but there are two star signals obscuring it. So, we now wish to use the algorithm derived in this section to subtract the stars from each individual image prior to stacking.

Table 4.6: Simulated scenario camera parameters.

Image Resolution	512 × 512
Field of View	1°
Focal Length	500 mm
Integration Time	3 s

Thus far we have discussed the process of building and fitting template stars to raw, unresolved imagery. What we have not looked at yet, however, is a method for placing the templates in the proper locations for the fitting process. The way that we perform template placement for the simulated case is through the use of a matched filter. We saw in Chapter 2 that a matched filter process is able to leverage prior knowledge of the appearance of a signal to detect that signal in an image even when it is below the noise floor [47]. Since we have a single template that should match all of the stars in the image, we may convolve it with the image to find locations that are likely to contain that signal. Figure 4.23 contains the matched filter convolution result for a single image in this sequence. The left-most image in the figure is the raw frame and the right-most image is the result of convolution with a single template star. The resulting image contains a peak for even dim star streaks.

One thing to note about the matched filter result in Figure 4.23 is that the edges contain additional artifacts. This is due to the fact that, as the template extends off the edge of the image, there is less data to compare to. This effect is an ongoing challenge for this star subtraction algorithm because it makes detection of edge-crossing streaks unreliable. For the simulation results discussed here, edge-crossing streaks are ignored and no attempt is made to subtract them. If, however, there was a way to precisely estimate the center point of an edge-cross streak, then it would be possible to subtract them with this algorithm.

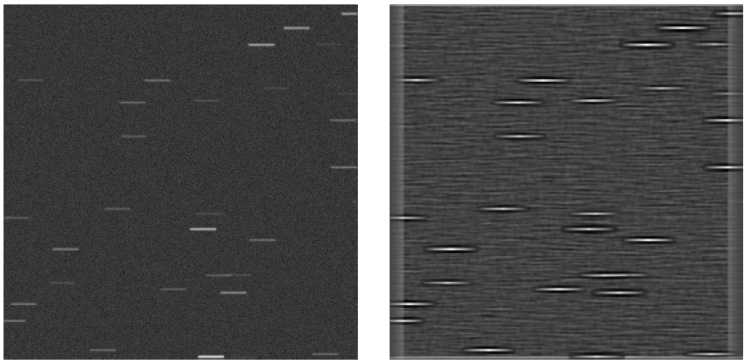

Figure 4.23: Result of a matched filter convolution process with a template star.

Another issue that arises from the matched filter process occurs when the signals of closely-spaced stars conflict. The left-most image in Figure 4.24 shows an example from the simulated scenario of this behavior. We see in the right-hand side of the figure that the result of matched filter process is corrupted due to the overlapping signals. The peak response corresponding to the left-most streak is correct, but the peak response for the right-most streak is shifted far to the right. In cases where multiple streaks overlap it is common to find unreliable location estimates for those signals.

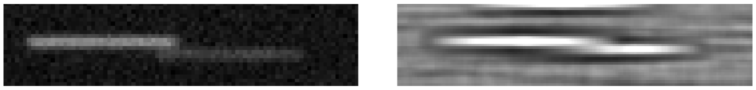

Figure 4.24: Matched filter convolution result with conflicting star signals.

Figure 4.19 shows the result of the star subtraction algorithm for this simulated scenario. The left-most image is the raw image stack that we saw before. The image on the right-hand side of the figure is the star-subtracted result. Except for the edge-crossing cases, all star streaks have been removed and the object of interest is clearly visible in the center of the image stack. Though there are some subtraction artifacts that arise in individual images, they end up buried in the noise and do not have a significant effect on the final result.

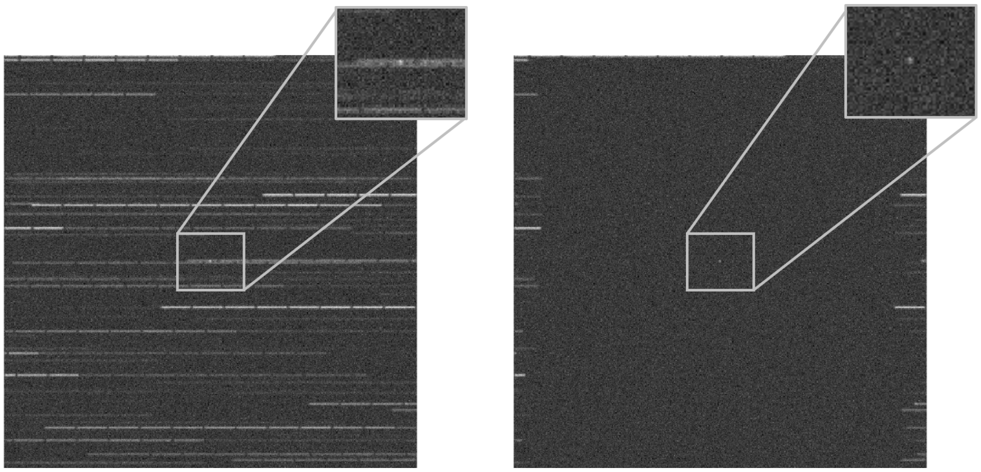

Figure 4.25: Simulated image stack (left) and the sum of the star-subtracted frames.

A key aspect of this algorithm to be aware of is that, due to the way that the optimization problem is structured, no signal of interest can be representable as a linear combination of the star templates. If this is the case, then there is a risk of subtracting the object of interest along with the star signals. For example, if all of the signals in the image (both stars and non-star objects) are point sources, then this approach will work to remove all of them equally. So, this method of star removal works best when the objects of interest look significantly different from the background star field. Of course, if one can discriminate between the stars and objects of interest beforehand, then it is possible to place the templates only on the star signals and preserve the target signals in some cases.

This star subtraction routine provides a global subtraction method that does not require an expensive nonlinear fit like the approach in [96] and does not risk corrupting the data further like the deblending technique in [40]. The optimization problems formulated for this problem are high dimensional problems, though. The dimension of the image vectors is mn and the template vector is $mn \times \ell$ so even small images require the handling of a large amount of data. Since each template contains only a single star, however, the A matrix is very sparse. By leveraging the sparsity of the problem and only operating on non-zero rows of A we can avoid memory shortages and some of the concerns about the computational cost of handling large matrices.

4.4 Summary

The work in this Chapter aims to make use of the dynamic motion models derived in Sections 3.1 and 3.2 to develop a more general set of data reduction approaches than those that currently exist. Though we have only focused on two motion models here (Equations 4.1 and 4.7), a similar process may be built around any of the equations of motion in Chapter 3. In fact, we may even develop more complicated models to account for specific behaviors of the sensor and space objects. In this way we can enable a wider variety of use cases for our data processing algorithms. This work is a fundamental step in the direction of a generalized data reduction pipeline for a diverse set of optical observers.

Section 4.1 detailed a fundamental approach to space object tracking and classification based on two image plane motion models. These algorithms enable a rigorous approach to analyzing space objects in optical imagery that can be completely driven by the data in the incoming imagery. Section 4.2 demonstrated an extension to traditional shift-and-add processes made possible by the motion models in Chapter 3. With this new approach it is possible to account for a wider range of motion profiles to extract a signal gain. Finally, section 4.3 demonstrated the use of this dynamic analysis to develop a template-based star subtraction algorithm. By predicting the motion of stars across the image plane we are able to build a template signal that can be used to subtract star clutter from images. This result allows us to remove undesired signals leaving only the space objects of interest.

In the context of the algorithms proposed in this chapter it is important to recall the limitations of the underlying dynamic models. The basic assumptions are that the optical sensor matches a particular model (the pinhole model in the implementation here) and the sensor undergoes a constant angular rotation rate. In addition, due to the choice of Equations 4.1 and 4.7 to describe the motion of objects in the image plane, we also must enforce a small field of view constraint. The particular choice of field of view depends on the acceptable error levels for the resulting data output. Since the algorithms here may be implemented with any of the dynamic motion models from Chapter 3, a number of other constraints may also apply. See the section in Chapter 3 corresponding to a particular motion model for more details on the necessary assumptions.

Chapter 5

Implications of the Geometric Analysis

This chapter will examine the possible applications of the geometric approach to image analysis. Chapter 3 considered two methods for geometrically analyzing optical sensors. The first method considered the relative geometry between single sensor rotating about a fixed axis and a space object. The resulting derivations showed that inertially-fixed objects move along paths described by conic sections. It is possible to leverage this knowledge for a number of applications. The first section in this Chapter will use this knowledge of star paths in an image to attempt to remove star streak curvature caused by imperfect sensor tasking. Further, we will see that it is possible to leverage this algorithm to augment the star subtraction approach derived in Chapter 4, so that it allows for global star removal even with curved star streaks. The second method of geometric analysis from Chapter 3 looked at the relationship among sensors in multi-sensor systems. That section showed that the it is possible to analytically project the line of sight vector describing a space object relative to a secondary sensor into a primary sensor of interest. The third section in this Chapter will leverage this geometry in order to enhance the non-star object discrimination capabilities of optical sensors. In addition, this approach has the benefit of immediately providing a range estimate to augment to the angle data that is already available.

5.1 Sensor Drift Corrections

The first application of the geometric approach to image analysis aims to improve the results of image processing techniques in cases where the sensor exhibits some undesired drift around the boresight during imaging. Measurements of unresolved signals in optical imagery are typically quantified by their centroids. See Chapter 2 for an in-depth discussion on the centroiding process. The accuracy of centroids typically decreases as point source signals move across the image plane during the exposure time [48]. The introduction of additional pixels – and by extension, additional noise – to the centroid process, the localization measurement becomes more variable. Further, drift around the boresight adds to this effect, corrupting streaks by causing them to deviate from an ideal straight-line path [78]. The resulting curvature induces a bias in the centroid result.

Figure 5.1 illustrates the effect of drift around the boresight on a streaked star signal. In this example, a star streak exhibits significant curvature. Though this level of curvature is not typical of real-world scenarios, it serves to visualize the centroid bias more easily. The centroid is marked by the '×' symbol. We see that the, because the centroid is essentially the average position of all of the pixels making up the signal, the centroid in this case ends up fall outside of the signal entirely. As the curvature increases, the average position moves further outward, until the signal makes a complete circle and the centroid corresponds to the center of that circle.

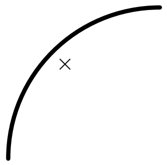

Figure 5.1: Illustration of a biased centroid (marked by the '×') of a curved streak.

Precise centroids are required for a number of different space situational awareness goals. Star centroids in particular provide attitude feedback [17], angular rate information [49], and sensor calibration parameters [18]. The work in this section will focus primarily on a geometric approach to removing the centroid bias from star signals. In Chapter 3, we saw that stars viewed by an observer rotating about a constant angular rate follow paths described by conic sections across the image plane. In addition, dynamic analysis showed that these paths are approximately described by circles centered on the center of rotation (Equation 3.67) for a small field of view sensor. This section will leverage both of these results and manipulate star images in a way which removes the curvature of the stars. These corrected images enable one to perform more traditional image reduction techniques even when the curvature of star streaks is severe.

5.1.1 Estimating the Axis of Rotation

In order to reduce the effect of a rotation during the exposure time, we first must determine some of the characteristics of that rotation. We will continue with the assumption that the sensor rotates about a fixed axis of rotation. Past derivations showed that, for a sensor rotating about a fixed axis, stars approximately move along circular paths centered around the center of rotation, (x_c, y_c) 3.75. Conveniently, it is often possible to estimate the center of rotation point completely from the data available in the image.

As few as two streaked stars are enough to derive the center of rotation. The fact that the stars must be streaked is an important one, because this derivation will rely on the assumption that it is possible to extract two distinct endpoints from a single streak. With two sets of endpoints, we can derive equations for the bisecting lines of each endpoint pair. The complete set of bisecting lines, in the ideal case, will intersect at the center of rotation. Figure 5.2 illustrates this geometry. Gray, dashed lines indicate the bisecting line for each star streak. The illustrated case assumes that the sensor is rotating entirely around the boresight, implying that the center of rotation is coincident with the origin of the focal plane coordinate system.

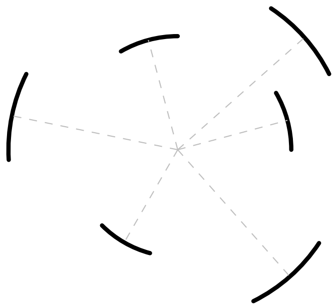

Figure 5.2: Bisecting lines of several star streaks intersecting at the center of rotation.

In the case of pure rotation around the boresight, the solution from Equation 3.60 is independent of the small field of view assumption. So, in this case, the stars do move in circles around the center of rotation. As the center of rotation deviates from the origin (ω_1 and ω_2 become larger), the neglected terms of 3.22 and 3.25 become more significant. The accuracy of this approach to estimating the center of rotation depends on the magnitudes of the angular rates ω_1 and ω_2.

Now, in order to formulate this problem numerically, consider again the normal form representation of a line. The normal form will be beneficial here because it will avoid the numerical issues arising for vertical or nearly vertical lines. The normal form equation is parameterized in terms of the minimum distance between the line and the origin, d, and the angle of the minimum distance line segment relative to the x axis, ν. See Figure 3.9 for an illustration of the normal form line parameterization. Recall that the standard normal form line is

$$d = x \cos\nu + y \sin\nu \tag{5.1}$$

To determine the bisecting line of a streak, assume that we have two unique endpoints given by the focal plane coordinates (x_1, y_1) and (x_2, y_2). Past work has shown that it is possible to extract endpoints of star streaks in some cases [78, 46]. We know that the bisecting line will be orthogonal to the line connecting the two endpoints and will pass through the midpoint. So, from Equation 5.1, we have two equivalent equations – one for each endpoint.

$$x_1 \cos\nu_{\|} + y_1 \sin\nu_{\|} = x_2 \cos\nu_{\|} + y_2 \sin\nu_{\|} \tag{5.2}$$

Here, $\nu_{\|}$ denotes the angle parameter of the line passing through the two endpoints. Solving the equation above for $\nu_{\|}$ yields

$$\nu_{\|} = \tan^{-1}\left(\frac{x_2 - x_1}{y_1 - y_2}\right) \tag{5.3}$$

This angle describes the line passing through the two endpoints, but we are interested in the line bisecting the two endpoints. The value of ν describing a line orthogonal to the one described by $\nu_{\|}$ is simply

$$\nu = \frac{\pi}{2} + \nu_{\|} = \frac{\pi}{2} + \tan^{-1}\left(\frac{x_2 - x_1}{y_1 - y_2}\right) \tag{5.4}$$

The midpoint of the endpoint pair, denoted by (x_m, y_m), is

$$x_m = \frac{x_1 + x_2}{2}, \qquad y_m = \frac{y_1 + y_2}{2} \tag{5.5}$$

So, Equation 5.4 gives the angle of the bisecting line. Knowing that the bisecting line passes through the midpoint, Equation 5.1 provides a solution for the parameter d.

$$d = x_m \cos\nu + y_m \sin\nu \tag{5.6}$$

Thus we have derived an equation describing the bisecting line of a single endpoint pair. Now, define ν_i and d_i for $i = 1, \ldots, n$, where n is the total number of endpoint pairs. These two terms represent the angle parameter and distance parameter for the ith endpoint pair, respectively. If our previous assumption is correct and the center of rotation (x_c, y_c) lies on each one of these lines, then the following system of equations describes describes each line evaluated at that point.

$$\begin{bmatrix} \cos\nu_1 & \sin\nu_1 \\ \vdots & \vdots \\ \cos\nu_n & \sin\nu_n \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \tag{5.7}$$

This system is over determined, but it is still possible to find the minimum least-squared error solution. The least-squares solution for (x_c, y_c) is

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \left(\begin{bmatrix} \cos\nu_1 & \sin\nu_1 \\ \vdots & \vdots \\ \cos\nu_n & \sin\nu_n \end{bmatrix}^T \begin{bmatrix} \cos\nu_1 & \sin\nu_1 \\ \vdots & \vdots \\ \cos\nu_n & \sin\nu_n \end{bmatrix} \right)^{-1} \begin{bmatrix} \cos\nu_1 & \sin\nu_1 \\ \vdots & \vdots \\ \cos\nu_n & \sin\nu_n \end{bmatrix}^T \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \tag{5.8}$$

The image-plane coordinates of the center of rotation will be sufficient for the derivations in the next section. Since the center of rotation is closely related to the axis of rotation, however, we can take this analysis a step further. Recall that the center of rotation is defined as the projection of the axis of rotation through the pinhole model (see Equation 3.67). The inverse projection equation (Equation 2.6) gives the rotation axis in terms of the center of rotation.

$$\boldsymbol{a} \approx \frac{1}{\sqrt{x_c^2 + y_c^2 + f^2}} \begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} \tag{5.9}$$

Note that this is only an approximate solution, and as the axis of rotation deviates from perfect alignment with the boresight, the quality of this solution suffers.

Other techniques for deriving the axis of rotation in a data-driven approach work by estimating the axis directly. In [49], the authors worked to develop a "stellar gyroscope" algorithm that is capable of estimating the angular rate from a single streaked star image. They showed that it is possible to estimate the angular rate from only a single streak. The algorithm works by projecting the pixels that make up a streak onto the unit sphere. They then fit a spherical circle (a circle defined in $\mathbb{R}^3$) to those data points. The normal vector to the resulting fit provides a measurement of the axis of rotation. This method is capable of estimate the axis of rotation to an accuracy as low as 1 degree, but it has a high computational cost [49].

A possible solution that avoids some of the high computational cost of the algorithm detailed in [49] involves extraction of streak endpoints. Past work has shown that it is possible to extract the endpoints of streaks to a tenth of a pixel or better [78, 46]. From the endpoints of at least three star streaks, it is possible to extract two attitude solutions, one for each endpoint set [18]. Further, from these attitude solutions, the rotation over the duration of the exposure time is available by solving Wahba's problem (see Equation 4.2) [76], for which there are several optimal solutions [77]. It is important to note, however, that these approaches do suffer from the small angular displacement of observations captured by a small field of view sensor. As noted in [49], the angular rate of the sensor and field of view may prove prohibitive in the number of stars visible at any given time. In the case that less than three stars are visible in an image, solving for the axis of rotation becomes impossible with this approach [77]. Further, if the visible stars did not move sufficiently to enable the extraction of endpoints, this solution is also not viable.

Of course, it is always possible to measure the rotation axis directly through the use of an on-board gyroscope. From an angular rate measurement, the axis of rotation is simply

$$\boldsymbol{a} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|_2} \tag{5.10}$$

Alternatively, a star identification routine operating on successive frames is capable provide attitude estimates from which an angular rate estimate can be derived [17].

5.1.2 Polar Image Resampling

Now we will leverage the geometric analysis from Chapter 4 to remove the curvature from a streaked star image. Specifically, we will use the fact that stars move along conic section paths according to Equation 3.182. Equation 3.75 showed that the these conic sections may be approximated as circular for small field of view sensors. Let us begin with that assumption to develop a basic algorithm. First, consider the center of rotation definition from Equation 3.67

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = -f\frac{1}{\omega_3}\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{5.11}$$

The center of rotation is defined as the projection of the axis of rotation, $\boldsymbol{a}$, into the image plane. According to Equation 3.75, all stars approximately move along circular paths around this point. Recall that this equation is

$$\begin{bmatrix} \hat{x}(t) \\ \hat{y}(t) \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix}\begin{bmatrix} \hat{x}(t_0) \\ \hat{y}(t_0) \end{bmatrix} \tag{5.12}$$

Now, if we define a polar coordinate system centered on (x_c, y_c) by

$$\begin{aligned} \hat{x} &= d\cos\nu \\ \hat{y} &= d\sin\nu \end{aligned} \tag{5.13}$$

then Equation 5.12 becomes

$$\begin{bmatrix} d\cos\nu \\ d\sin\nu \end{bmatrix} = \begin{bmatrix} \cos(\omega_3 t) & \sin(\omega_3 t) \\ -\sin(\omega_3 t) & \cos(\omega_3 t) \end{bmatrix}\begin{bmatrix} d_0\cos\nu_0 \\ d_0\sin\nu_0 \end{bmatrix} \tag{5.14}$$

Consider the first equation of the system above.

$$d\cos\nu = d_0\cos(\omega_3 t)\cos\nu_0 + d_0\sin(\omega_3 t)\sin\nu_0 \tag{5.15}$$

By trigonometric identities, the right-hand side is equivalent to

$$d\cos\nu = d_0\cos(\omega_3 t - \nu_0) \tag{5.16}$$

The second equation is

$$d\sin\nu = -d_0\sin(\omega_3 t)\cos\nu_0 + d_0\cos(\omega_3 t)\sin\nu_0 \tag{5.17}$$

which similarly simplifies to

$$d\sin\nu = d_0\sin(\nu_0 - \omega_3 t) \tag{5.18}$$

Solving the first equation for d yields

$$d = d_0\frac{\cos(\omega_3 t - \nu_0)}{\cos\nu} \tag{5.19}$$

and inserting this solution into Equation 5.18 shows that

$$d_0\cos(\omega_3 t - \nu_0)\frac{\sin\nu}{\cos\nu} = d_0\sin(\nu_0 - \omega_3 t) \tag{5.20}$$

Notice that the d_0 term on both sides cancels out. Further, separating like terms leads to

$$\frac{\sin\nu}{\cos\nu} = \frac{\sin(\nu_0 - \omega_3 t)}{\cos(\omega_3 t - \nu_0)} \tag{5.21}$$

Since $\cos(\beta) = \cos(-\beta)$ for any β, the relation above is equivalent to

$$\frac{\sin\nu}{\cos\nu} = \frac{\sin(\nu_0 - \omega_3 t)}{\cos(\nu_0 - \omega_3 t)} \tag{5.22}$$

So, we see that ν is simply

$$\nu = \nu_0 - \omega_3 t \tag{5.23}$$

Inserting this back into the original solution for d shows that

$$d = d_0 \frac{\cos(\omega_3 t - \nu_0)}{\cos(\nu_0 - \omega_3 t)} = d_0 \tag{5.24}$$

So, in a polar coordinate system centered on the center of rotation, according to the approximate motion model from Equation 3.60, inertially fixed objects remain at a fixed distance from the origin and the angle evolves proportionally to the angular rate around the boresight. This result corroborates the fact that stars in the simplified motion model are expected to move in circles around the center of rotation. Figure 5.3 illustrates this exact scenario. In the figure, a number of stars (indicated in black) rotate around the center of rotation over a relatively long exposure. The gray, dotted lines indicate the polar coordinate system.

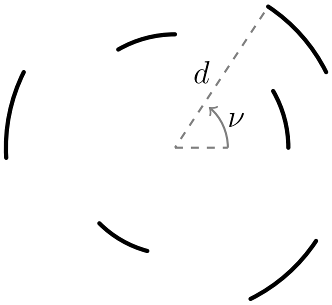

Figure 5.3: A polar coordinate system placed at the center of rotation.

An alternative to sampling in a polar coordinate system uses the full solution from Equation 3.182. Since the solution above relies on the approximate equations of motion, the neglected terms may lead to some residual curvature. To fully negate this curvature, it is possible to instead resample along a conic section path parameterized by that curve's minimum distance from the center of rotation. The angle parameter from Equation 5.25 may be defined in the same way, as the angle of the current point with respect to the x-axis, but the relationship between the angular rate of the sensor and the rate of change of ν is complicated. An analytic solution for this relationship is still unknown. For now, we will only consider the polar coordinate approach.

Let us consider some simulated scenarios to examine the effects of processing an image in a polar coordinate system. Define the $\mathbb{R}^2 \to \mathbb{R}^2$ transformation from focal plane coordinates to polar coordinates as

$$\begin{aligned} x &= d\cos\nu + x_c \\ y &= d\sin\nu + y_c \end{aligned} \tag{5.25}$$

Figure 5.4 contains the first example case that we will consider. In this case, the sensor is viewing a only stars. During imaging, the sensor rotates by approximately 10 degrees about the boresight direction, causing the stars to streak. From Equations 5.11 5.12, we know that the stars are moving in circular paths centered on the center of rotation. Further, the center of rotation is coincident with the origin of the focal plane coordinate system (See Figure 2.5 for the definition of the coordinate system). A small amount of noise corrupts each pixel's intensity value.

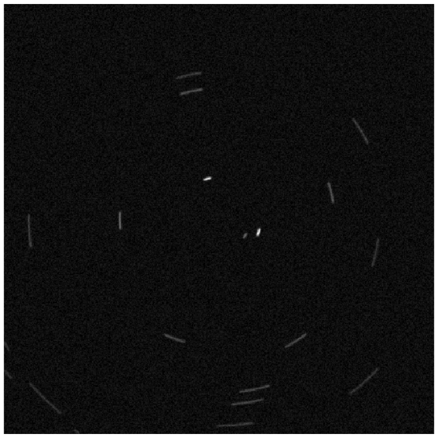

Figure 5.4: Simulated star image with pure rotation around the boresight.

The image above is seen again in Figure 5.5, expressed in the polar coordinate system. The polar coordinate system from Equation 5.25 with sampling bounds $d = 1, 2, \ldots, d_{\text{diag}}$, where d_{diag} is the distance from the center of the image to an of the four corners, and $\nu = 0°, 2°, \ldots, 360°$. For each point in a (d, ν) grid, Equation 5.25 provides the corresponding (x, y) coordinate. This set of focal plane coordinates then allows us to compute a value for that location in the resampled image. The results in Figure 5.25 used a linear interpolation amongst nearby pixels for each coordinate in the grid.

There are a few key caveats of this algorithm which are visible as artifacts in Figure 5.5. The first issue affects pixels which are close to the center of rotation. As the distance of a particular pixel from the center of rotation becomes smaller, that pixel is sampled more densely. This oversampling effectively stretches the signal across multiple pixels in the resampled image. The artifacts due to this are visible near the top of the image in Figure 5.5. Though there are no star signals in this region of the Figure, oversampling of the background noise causes correlations between pixels that did not exist before. This effect can lead to spurious detections, so special care should be taken in the region near the center of rotation. Ideally, a minimum bound for d should be chosen such that no single pixel is sampled more than once.

Another effect seen in Figure 5.5 is toward the bottom of the image, corresponding to larger distances from the center of rotation. Each of these periodically-occurring artifacts corresponds to the corner of the original image. In order to fully sample a square image with a circular coordinate system, the grid must sometimes extend beyond the bounds of the image frame. Since there is no data for what exists outside of the image borders, these data points should simply be marked as invalid and not considered in later processing. Special care must be taken around these regions as they may also lead to spurious object detections.

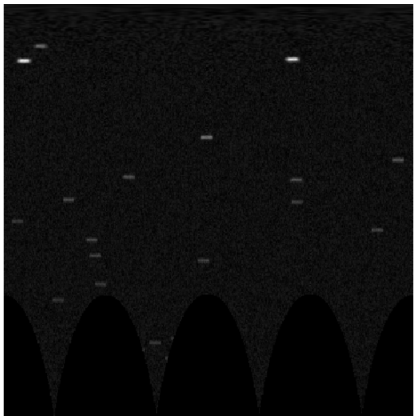

Figure 5.5: Curved star streak image resampled in a polar coordinate system.

Another concern when resampling the image data is the choice of sampling bounds. For this case, the choice of bounds was simple, since the center of rotation was in the center of the focal plane. In the general case, however, where the center of rotation can be anywhere in the plane of the CCD (including well beyond the borders of the image data), the choice of sampling bounds can be much more complicated. A simple method for solving this problem is to consider the location of each of the pixels on the border of the image as projected into the polar coordinate system. From 5.25, it follows that the distance parameter in polar coordinates for any (x, y) location in the focal plane is

$$d = \sqrt{\left(x - x_c\right)^2 + \left(y - y_c\right)^2} \tag{5.26}$$

So, if we project n pixel locations along the edge of the focal plane into the polar coordinate system, the choice of bounds is simply

$$\begin{aligned} d_{\min} &= \min\left\{d_1, d_2, \ldots, d_n\right\} \\ d_{\max} &= \max\left\{d_1, d_2, \ldots, d_n\right\} \end{aligned} \tag{5.27}$$

Note that this is not an exact solution and may lead to up to half a pixel of error. Increasing the maximum distance and decreasing the minimum distance by half of the size of a pixel may cause some additional out-of-bounds sampling, but it will avoid the loss of any image data. The angle, ν, in terms of an arbitrary (x, y) coordinate and the center of rotation, (x_c, y_c) is

$$\nu = \tan^{-1}\left(\frac{y - y_c}{x - x_c}\right) \tag{5.28}$$

A quadrant check is required to determine the appropriate value of ν. The proper choice of bounds for ν is

$$\begin{aligned} \nu_{\min} &= \min\left\{\nu_1, \nu_2, \ldots, \nu_n\right\} \\ \nu_{\max} &= \max\left\{\nu_1, \nu_2, \ldots, \nu_n\right\} \end{aligned} \tag{5.29}$$

except in the case where the center of rotation is within the bounds of the original image. When the center of rotation is within the original image data, then the sampling bounds should always span a full rotation.

It is worth noting that this technique for image analysis is not limited to directly manipulating the image intensity data. In fact, it is possible to perform analysis based on the polar coordinate system without ever manipulating the image data. For example, Figure 5.1 highlights a key problem of analyzing data with curved streaks. If, instead we convert the coordinates of each pixel to their polar-coordinate counterparts, then the centroid of an object may be written as

$$\begin{bmatrix} c_d \\ c_\nu \end{bmatrix} = \sum_{i=1}^{n} w_i \begin{bmatrix} d_i \\ \nu_i \end{bmatrix} \tag{5.30}$$

where w_i is the intensity-based weighting of the ith pixel and (d_i, ν_i) describes the location of that pixel in the polar coordinate system. So, the image intensity data remains the same, but rather the coordinates for each pixel are modified prior to the traditional centroid calculation. The resulting centroid does not contain the same bias seen in Figure 5.1. As seen in Figure 5.5 and from the results in Equations 5.23 and 5.24, in the new coordinate system the streaks have no curvature. Thus the bias due to curvature does note affect results computed in the polar coordinates.

Issues can arise here due to the fact that the angle, ν_i, is discontinuous at one full rotation. In cases where a streak extends over the boundary between 2π and 0 radians, it is necessary to shift the coordinates so that it does not fall on this boundary prior to computing the centroid with Equation 5.30. So, from Equation 5.25, the corresponding centroid in the original cartesian coordinate space is

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} c_d \cos c_\nu + x_c \\ c_d \sin c_\nu + y_c \end{bmatrix} \tag{5.31}$$

Thus, it is possible to derive an unbiased centroid from a curved streak without modifying the original image intensity values directly.

5.1.3 Spherical Image Resampling

The process from the previous section has the limitation that it relies on the approximate motion models derived in Chapter 3. A more general approach relies on unit vectors describing points on the image plane of the sensor projected back onto the unit sphere. A resampling procedure in a spherical coordinate system, then, should provide a similar result as we saw in the previous section. Consider a spherical coordinate system chosen in the following way:

$$\boldsymbol{u} = \begin{bmatrix} \sin\alpha\cos\beta \\ \sin\alpha\sin\beta \\ \cos\alpha \end{bmatrix} \tag{5.32}$$

Note that here we neglect the distance from the origin. The following algorithm relies entirely on unit vector descriptions of space objects, for which the distance from the origin is always 1. The change of coordinate systems in the equation above is not ideal, because it is completely independent of the axis of rotation. Instead, it is desirable to relate the orientation of the spherical coordinate system to the axis of rotation. Consider an alternate definition of the coordinate system

$$\boldsymbol{u'} = R_a^T\boldsymbol{u} = \begin{bmatrix} \sin\alpha\cos\beta \\ \sin\alpha\sin\beta \\ \cos\alpha \end{bmatrix} \tag{5.33}$$

where R_a describes a rotation that aligns the pointing vector of the sensor with the axis of rotation and $\boldsymbol{u'}$ is a particular unit vector expressed in this new reference frame. So, in this new formulation of the spherical coordinate system, the line defined by $\alpha = 0$ corresponds to the axis of rotation. In a moment we will see the benefits of this particular choice of coordinate system. First, let us derive the rotation matrix, R_a. Define a vector, $\boldsymbol{a}_\perp$, by

$$\boldsymbol{a}_\perp = \hat{\boldsymbol{e}}_3 \times \boldsymbol{a} \tag{5.34}$$

This vector is orthogonal to both the pointing vector of the sensor, $\hat{\boldsymbol{e}}_3$ in the sensor's reference frame, and the axis of rotation, $\boldsymbol{a}$. The angle between these two vectors is

$$\theta_\perp = cos^{-1}\left(\hat{\boldsymbol{e}}_3 \cdot \boldsymbol{a}\right) \tag{5.35}$$

So, a rotation matrix that aligns $\hat{\boldsymbol{e}}_3$ with $\boldsymbol{a}$ is

$$R_a = I - \tilde{\boldsymbol{a}}_\perp \sin\theta_\perp + \tilde{\boldsymbol{a}}_\perp^2\left(1 - \cos\theta_\perp\right) \tag{5.36}$$

Though this definition of rotation matrix is not unique, different choices of R_a will not effect the final result. Now, the evolution of the $\boldsymbol{u}'$ unit vector in the with respect to time in the axis-aligned coordinate system is simply [8]

$$\boldsymbol{u}'(t) = \begin{bmatrix} \cos\left(\|\boldsymbol{\omega}\|\, t\right) & -\sin\left(\|\boldsymbol{\omega}\|\, t\right) & 0 \\ \sin\left(\|\boldsymbol{\omega}\|\, t\right) & \cos\left(\|\boldsymbol{\omega}\|\, t\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{u}'(t_0) \tag{5.37}$$

where the norm is the 2-norm. In the axis-aligned reference frame, any unit vector $\boldsymbol{u}$ revolves around the vertical axis with a rate equivalent to the magnitude of the angular rate vector in the non-aligned reference frame. The individual components of the relationship above are

$$\begin{aligned} u_1'(t) &= u_1'(t_0)\cos(\|\boldsymbol{\omega}\| t) - u_2'(t_0)\sin(\|\boldsymbol{\omega}\| t) \\ u_2'(t) &= u_1'(t_0)\sin(\|\boldsymbol{\omega}\| t) + u_2'(t_0)\cos(\|\boldsymbol{\omega}\| t) \\ u_3'(t) &= u_3'(t_0) \end{aligned} \tag{5.38}$$

Define α_0 as the value of α corresponding to $\boldsymbol{u}'(t_0)$. Similarly, choose β_0 to represent the value of β for $\boldsymbol{u}'(t_0)$. From Equation 5.33, we find

$$\cos\alpha = \cos\alpha_0 \tag{5.39}$$

which implies that

$$\alpha = \alpha_0 \tag{5.40}$$

In a spherical coordinate system aligned with the axis of rotation of a sensor, any unit vector rotating about that axis maintains a constant value for α. From the $\boldsymbol{u}'$ term of Equation 5.38 and the spherical coordinate system definition in Equation 5.33, we have

$$\sin\alpha\cos\beta = \sin\alpha_0\cos\beta_0\cos(\|\boldsymbol{\omega}\| t) - \sin\alpha_0\sin\beta_0\sin(\|\boldsymbol{\omega}\| t) \tag{5.41}$$

Since $\alpha = \alpha_0$,

$$\cos\beta = \cos\beta_0\cos(\|\boldsymbol{\omega}\| t) - \sin\beta_0\sin(\|\boldsymbol{\omega}\| t) \tag{5.42}$$

By trigonometric identities, the right hand side reduces such that

$$\cos\beta = \cos(\beta_0 + \|\boldsymbol{\omega}\| t) \tag{5.43}$$

Finally, the equation above implies that

$$\beta = \beta_0 + \|\boldsymbol{\omega}\| t \tag{5.44}$$

So, in a polar coordinate system where the axis of rotation is aligned with the line described by $\alpha = 0$, the angle α of an object rotating about that axis is constant and the corresponding β value evolves according to the magnitude of the angular rate. This result is similar to that of the polar coordinate system case and in fact we are able to leverage it in the same way to remove the curvature of imaged star streaks.

Once again we will consider a simulated case to determine the efficacy of a spherical resampling process. Figure 5.6 contains a simulated star image form a rotating platform. The exposure time and rotation rate are cause significant length in the imaged stars. Note that the stars in this image are much more straight than those in Figure 5.4. In fact, since the spherical process described here does not depend on the center of rotation, it continues to work even when the center of rotation is undefined. This scenario occurs when there is no rotation around the boresight over the exposure time (i.e. $\omega_3 = 0$). The spherical technique

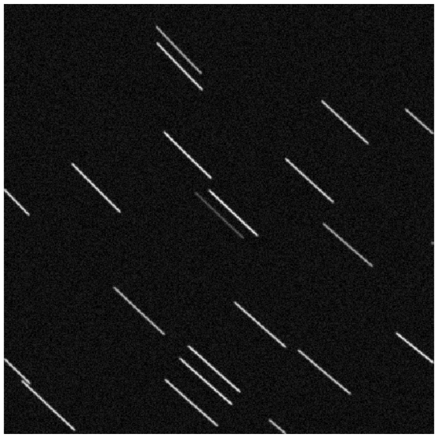

Figure 5.6: Simulated star streak image with limited rotation around the boresight.

is only concerned with the description of the axis of rotation in $\mathbb{R}^3$, so it function for any arbitrary angular rate vector.

The image resulting from a spherical resampling of Figure 5.6 is seen in Figure 5.7. This case sampled the full celestial sphere ($\alpha \in [0, 2\pi]$ and $\beta \in [0, \pi]$) and removed extraneous data while preserving a rectangular image shape. This grid in (α, β) leads to a grid of unit vectors from Equation 5.33 and further a grid of image plane points through the $\mathbb{R}^3 \to \mathbb{R}^2$ projection. In this case, we consider the pinhole model (Equation 2.4), but any other projection model is valid as well. A linear interpolation provided intensity values for each grid point based on the neighboring pixel intensities.

As in the case of the polar resampling, we again see that it is often necessary to sample points that are outside the boundary of the original image. The dark regions around the corners of the image in Figure 5.7 are points that were sampled from out of bounds. These points contribute no data, but are required to produce a rectangular image. In practice, these pixels should be marked as invalid and ignored in subsequent processing. The spherical approach is also limited in cases where the axis of rotation intersects the image plane, though that is not the case in Figure 5.6. In the region around $\alpha = 0$ pixels tend to become heavily oversampled, stretching their signal out over many pixels in the resampled space. Limits for α should be chosen carefully to avoid oversampling noise and creating spurious detections.

Now, we would like to determine the minimal sampling bounds for an arbitrary axis of rotation. From Equation 5.33, we may find that α in terms of the unit vector it describes is

$$\alpha = \cos^{-1}(u_3) \tag{5.45}$$

and β is

$$\beta = \tan^{-1}\left(\frac{u_2}{u_1}\right) \tag{5.46}$$

So, we can proceed in the same way as for the polar resampling bounds. The projection of

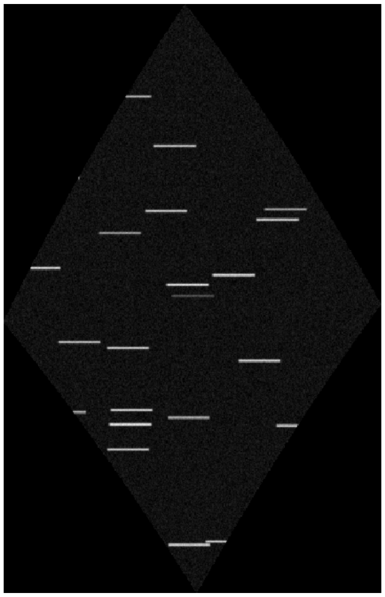

Figure 5.7: Star streak image resampled in a spherical coordinate system.

all of the edge points in the original image with knowledge of the axis of rotation provides a data set from which we can extract the minimal sampling bounds. First, project each (x, y) coordinate through any desired $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ transformation to find a corresponding $\boldsymbol{u}$. Then, use the two equations above to compute the corresponding (α, β) coordinate. The minimum and maximum values for α are

$$\begin{aligned} \alpha_{\text{min}} &= \min\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \\ \alpha_{\text{max}} &= \max\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \end{aligned} \tag{5.47}$$

and the minimum and maximum values for β are

$$\begin{aligned} \beta_{\text{min}} &= \min\{\beta_1, \beta_2, \ldots, \beta_n\} \\ \beta_{\text{max}} &= \max\{\beta_1, \beta_2, \ldots, \beta_n\} \end{aligned} \tag{5.48}$$

Though this way of determining the bounds requires the projection of a potentially larger number of points, it greatly reduces the computational cost of interpolating a large number of unnecessary values. In the case where the axis of rotation passes through the image, this analysis is made somewhat more simple. In that case, the β term must complete one full rotation to completely capture the image data.

Similar to the polar case, the spherical approach is not limited to analysis which directly manipulates the image data. For example, it is also possible to compute centroids for objects in the spherical coordinate system without resampling the original image. Instead, we are able to modify the pixel coordinates while retaining the original intensity data. The centroid of an object in spherical coordinates is

$$\begin{bmatrix} c_\alpha \\ c_\beta \end{bmatrix} = \sum_{i=1}^{n} w_i \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \tag{5.49}$$

where w_i is the intensity-based weighting of the ith pixel and (α_i, β_i) describes the location of that pixel in the spherical coordinate system. Issues can arise here due to the fact that both angles are discontinuous. In cases where a streak extends over the boundary for either α or β, it is necessary to shift the coordinates so that it does not fall on this boundary prior to computing the centroid with Equation 5.49. So, from Equation 5.33, the corresponding unit vector describing the centroid in the axis-aligned cartesian coordinate space is

$$\begin{bmatrix} c'_1 \\ c'_2 \\ c'_3 \end{bmatrix} = \begin{bmatrix} \sin c_\alpha \cos c_\beta \\ \sin c_\alpha \sin c_\beta \\ \cos c_\alpha \end{bmatrix} \tag{5.50}$$

Recall the relationship between the axis-aligned coordinate system and the sensor's original body-fixed reference frame. The centroid vector in the sensor's reference frame is related to the vector above by

$$\boldsymbol{c}' = R_a^T \boldsymbol{c} \tag{5.51}$$

where $\boldsymbol{c}' = (c'_1, c'_2, c'_3)$ and $\boldsymbol{c} = (c_1, c_2, c_3)$. Typically, the unit vector $\boldsymbol{c}$ is the desired result. If necessary, the centroid coordinate in the image plane is given through whichever projection equation most accurately describes the sensor. For a model adhering to the pinhole model, the centroid coordinate (c_x, c_y) is

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = -\frac{f}{c_3} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{5.52}$$

Thus, the spherical analysis provides bias-free centroid measurements without requiring direct manipulation of the image data.

5.2 Star Subtraction with Field Curvature

Analyzing imagery in a polar coordinate system that originates at the center of rotation also benefits the star removal algorithm derived in Chapter 4. Recall that the star removal algorithm relied on the motion model in Equation 3.62, which assumed no rotation around the boresight of the sensor (i.e. $\omega_3 = 0$). Under this assumption, all of the star signals in an image are approximately equal and therefore the star removal process only requires a single template. If we first resample the image in a polar coordinate system originating at the center of rotation defined in Equation 3.67, then the equations of motion describing the motion of a star across the focal plane are

$$\begin{aligned} \nu(t) &= \nu(t_0) - \omega_3 t \\ d(t) &= d(t_0) \end{aligned} \tag{5.53}$$

See Figure 5.3 for a diagram of the (d, ν) polar coordinate system. So from the equations above we can infer that, in the polar coordinate system, all of the star signals will have approximately identical appearance. Thus, by resampling about the center of rotation, we are able to use a common template for all of the stars in an image, thereby enabling the star removal process for Chapter 4 even when $\omega_3 \neq 0$.

Figure 5.8 contains the simulated example case that will serve as a proof of concept for star subtraction of curved signals. In this scenario, the sensor is rotating entirely around the boresight of the sensor. According to Equation 3.60, the stars are tracing circles around the center of the image. The left-most image in Figure 5.8 contains the resulting raw image data. After resampling the image a in polar coordinate system about the center of rotation, we arrive at the image on the right-hand side of the figure. The vertical axis corresponds to the d parameter, increasing from zero at the top of the image. The horizontal axis corresponds to the ν parameter, beginning from zero and increasing from left to right.

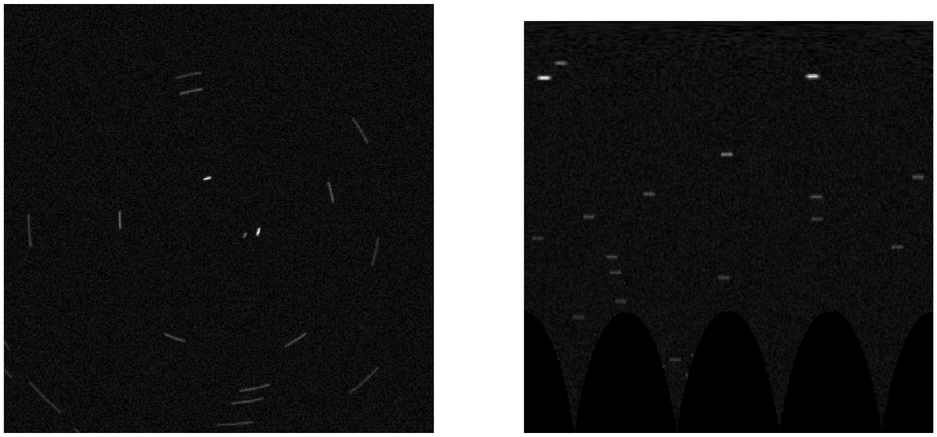

Figure 5.8: Simulated image with boresight rotation (left) and its polar resampling (right).

We follow the same process for star removal as we did in Chapter 4 for this case, with the difference that the fitting and subtraction process is performed in the resampled space. After fitting the stars in the resampled image, we transform the template image back into the original coordinate space and subtract the result. This algorithm run on the set of images above produced the results in Figure 5.9. The left-hand image in the figure is the polar resampled image from Figure 5.8 with the stars removed. Rather than mapping this image back into the original coordinates, we map the template image back and perform the subtraction on the raw image data. The result of this process is seen in the right-most image of Figure 5.9. We see that the algorithm was successful at removing the bulk of the star signals.

Something to note in the final image of Figure 5.9 is that there are a number of artifacts near the center and outer portions of the image. The artifacts near the center correspond to the upper portion of the resampled image, where pixels become “stretched” due to oversampling.

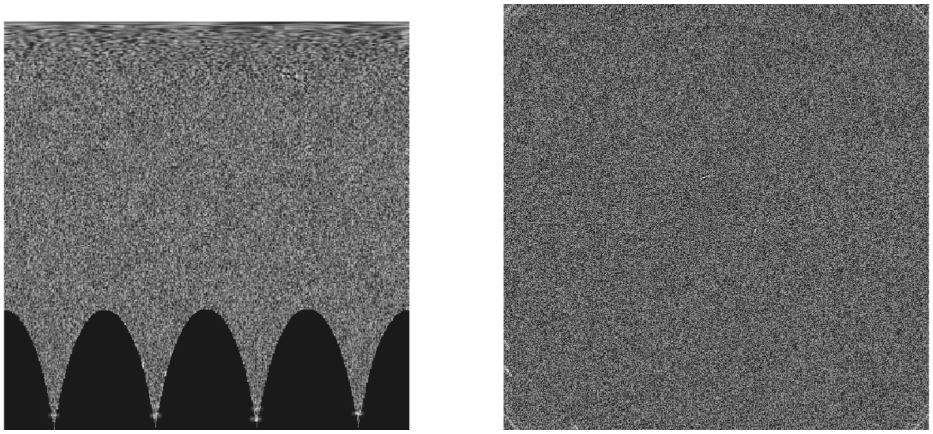

Figure 5.9: Star-subtracted result in polar coordinates (left) and original coordinates (right).

This causes correlation in the noise among pixels and can lead to problems in post-processing. In this case, the matched filter star extraction process finds a large number of spurious signals in that region. The artifacts near the corners of the image correspond to error introduced by the undefined regions near the bottom of the resampled image. Though these regions are ignored in processing, pixels on the border can be difficult to handle. Special care must be taken in this region as well to avoid spurious detections in the matched filter process.

Also of note is the error introduced by the interpolation. The results here use a linear interpolation among pixels to transform from the raw image data to the resampled space. Of course, some information is lost and distorted in this process. The result is that the templates, though theoretically correct, may not fit exactly to the underlying data. Due to this effect, it is best to avoid manipulating the raw image at all. Rather, using an algorithm that maps the template, defined in polar coordinates, into the original image space and performs the star subtraction without interpolation will produce a more accurate result.

5.3 Multi-Observer Space Object Discrimination

This section will leverage the epipolar analysis from Chapter 3 to enhance the detection capabilities of multi-sensor systems. We know that two observers with a sufficient baseline separation can also provide an immediate range estimate to a particular object if operated simultaneously [19]. This range estimate is only available if both sensor systems are confident that they are observing the same object. Since unresolved objects have nearly identical appearances in optical imagery, it is often non-trivial to associate objects between frames.

Prior work has examined the use of epipolar geometry for automatic detection and ranging of non-star objects. The work in [66] and [67] describes an algorithm that leverages the epipolar geometry of a wide baseline stereo system to detect space objects. This technique works by transforming the image from one sensor so that the stellar background matches that of the secondary sensor. Then, any signal which has moved is likely to be a non-star object of interest [66]. This algorithm has been demonstrated on a real-world stereo sensor system with a 37 km baseline and was able to detect six satellites in a three hour period in MEO and Molniya orbits [66].

The work here aims to extend and generalize the multi-sensor approach to space object detection and ranging. From a single stereo pair there is the possibility for an ambiguity in the matching process. We know that an object detected in a primary sensor will lie along the corresponding epipolar line in the secondary sensor, but star clutter and other space objects may also cross this line. In this case, there is an additional difficulty in determining the correct sensor-to-sensor object association. The work in [66] solves this by transforming the stellar background from one sensor to match that of the secondary sensor, but this process is not always possible. If there is no overlap in the two image frames, then this transformation will not provide any new information and will likely corrupt the existing data.

Figure 5.10 illustrates the stereo pair geometry. In this figure we see that a secondary sensor in the background observes a spacecraft and reports a corresponding line of sight vector. The primary sensor in the foreground images the spacecraft as well and artificially enhances the image by drawing the line of sight vector. The line crosses the spacecraft and there are no other objects in the image, so the association is simple. If, on the other hand, a star or other space object fell along this line, then the association would be ambiguous.

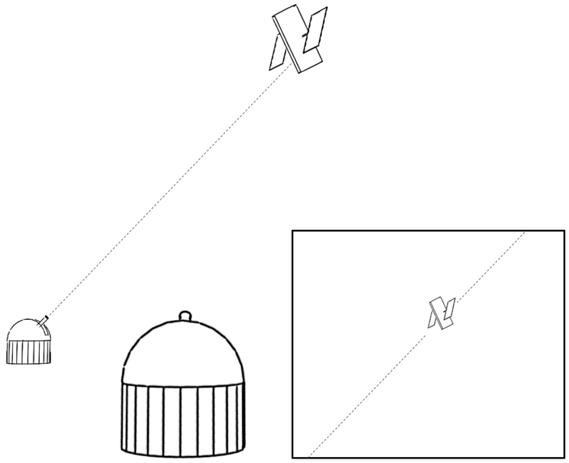

Figure 5.10: Diagram of the epipolar geometry between two optical observers.

Recall that the equation describing the line of sight vector from a secondary sensor projected into the primary sensor is

$$y = \frac{u_3\left(\boldsymbol{r}_o\right)_2 - u_2\left(\boldsymbol{r}_o\right)_3}{u_3\left(\boldsymbol{r}_o\right)_1 - u_1\left(\boldsymbol{r}_o\right)_3} x + f\frac{u_2\left(\boldsymbol{r}_o\right)_1 - u_1\left(\boldsymbol{r}_o\right)_2}{u_1\left(\boldsymbol{r}_o\right)_3 - u_3\left(\boldsymbol{r}_o\right)_1} \tag{5.54}$$

where $\boldsymbol{r}_o$ is the position of the secondary observer in the primary sensor's body-fixed reference frame and $\boldsymbol{u}$ is the unit vector describing the optical measurement of an object of interest from that sensor. The line of sight vector, $\boldsymbol{u}$, is also expressed in the reference frame of the primary sensor. Here, $(\boldsymbol{r}_o)_i$ denotes the ith element of the $\boldsymbol{r}_o$ vector. This relationship allows us to analyze imagery from multi-sensor systems in terms of the epipolar geometry between them.

Let us consider some simulated example cases to see the benefit of using more than two sensors simultaneously. The first simulated case, in Figure 5.11, contains only two objects and no stellar background. The image on the left-hand side of the figure contains only one of the two objects while the image on the right-hand side contains both. The blue lines across each image indicate the line of sight vectors from the other sensor, drawn according to Equation 5.54. The line of sight vector from the sensor on the left crosses one object in the image on the right, indicating that those two signals correspond to the same object.

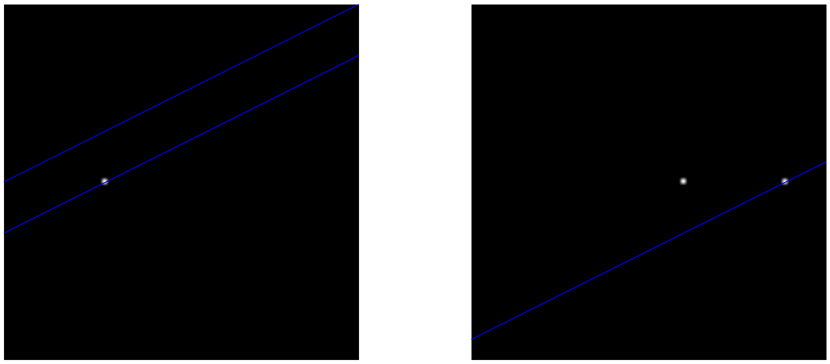

Figure 5.11: Simulated images from a stereo pair with epipolar line overlay.

Figure 5.12 shows a higher fidelity simulation that includes a star field from the Tycho-2 star catalog [6]. In this simulated case all three sensors are oriented in the same direction and each image contains a single space object. The epipolar lines overlaid on each image indicate the line of sight vectors from the other two sensors corresponding to that space object. We see that in each image, the line of sight vectors from the other two sensors cross at the image plane location of the object of interest. This follows intuitively from the geometry of the problem. We know that any observed object lies at some unknown distance along the line of sight vector. If we have two line of sight vectors from different locations for the same object, then the object must be at the intersection of these two lines.

Thus far we have considered only perfect line of sight measurements. In real-world scenarios there will be some error associated with each measurement and the line of sight vectors may not intersect exactly at the location of an object of interest. To account for this uncertainty, we represent the measurement error as a bounded rotation, E, from nominal epipolar line,

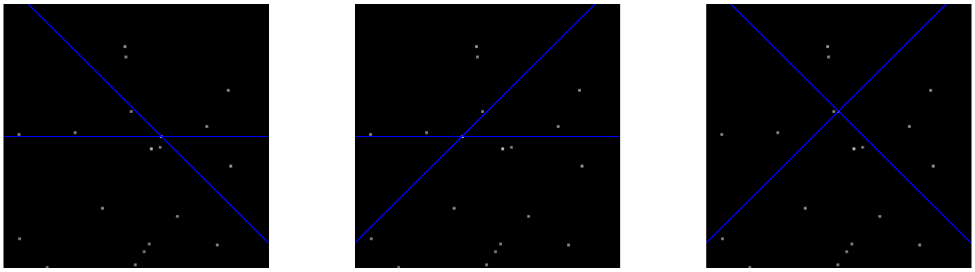

Figure 5.12: Epipolar lines for a three camera system architecture.

$\boldsymbol{u}_n$. The error bounds are

$$\boldsymbol{u}_e = E\boldsymbol{u}_n \tag{5.55}$$

where E is any rotation matrix with a principal angle that describes the maximum uncertainty in the line of sight vector. So, the correct line of sight vector may be any vector in a cone around the nominal line. We are in particular interested in the maximally-separated error bounds from the point of view of a secondary sensor. In other words, we desire the error bounds that, when projected into another sensor, are as widely separated as possible. These error bounds are of interest because they will provide the most conservative error bounds for subsequent analysis. Denote $\boldsymbol{r}_{i,j}$ as the relative position of the ith sensor (the sensor that measured the line of sight vector) to the jth sensor (the sensor in which we wish to project the line of sight vector). To produce the maximally separated error bounds we must only consider error bounds external to the $\boldsymbol{u}_n - -\boldsymbol{r}_{i,j}$ plane. Any rotation in this plane will move the line of sight closer to or further from the receiving sensor, causing no visible change from the nominal vector. The cross product $\boldsymbol{r}_{i,j} \times \boldsymbol{u}_n$ produces the normal to this plane. Any rotation around the nominal line, $\boldsymbol{u}_n$ will not produce any change in the resulting vector so the axis of rotation must also be orthogonal to this vector. So, the relation

$$\boldsymbol{a}_{\max} = \boldsymbol{u} \times (\boldsymbol{r}_{i,j} \times \boldsymbol{u}_n) \tag{5.56}$$

gives the axis of rotation that produces the maximally-separated error bounds. Note that $\boldsymbol{a}_{\max}$ must be normalized to produce a valid rotation axis. We have assumed that the rotation from the nominal value is bounded by some angle, denoted here by $\theta_{\max}$, so E is [8]

$$E = I - \tilde{\boldsymbol{a}}_{\max} \sin\theta_{\max} + \tilde{\boldsymbol{a}}_{\max}^2 (1 - \cos\theta_{\max}) \tag{5.57}$$

where I is the identity matrix and $\tilde{\boldsymbol{a}}_{\max}$ is the skew-symmetric form of $\boldsymbol{a}_{\max}$. So, the epipolar error bounds are

$$\begin{aligned} \boldsymbol{u}_+ &= E\boldsymbol{u}_n \\ \boldsymbol{u}_- &= E^T\boldsymbol{u}_n \end{aligned} \tag{5.58}$$

Figure 5.13 contains a simulated image of a space object with nominal epipolar lines and error bounds from two external sensors. The teal, dashed lines are the nominal line of sight vectors and the blue, solid lines are the epipolar error bounds. We see that the intersection of these two regions is a small, approximately 4 square pixel area that contains the object of interest. The fact that this intersection contains an objects suggests a strong likelihood that it should be associated with the corresponding signal in the other sensors.

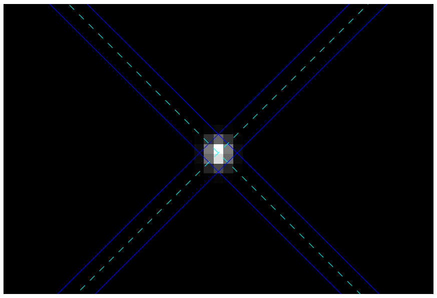

Figure 5.13: Epipolar lines from two external sensors with error bounds.

Figure 5.14 shows another simulated scenario, this images from all three sensors and the corresponding epipolar lines. Again, the intersection of the epipolar error bounds contains only a small region of the image which also contains an object in each image. It is this behavior that we can leverage to make sensor-to-sensor object associations. This effect scales up easily with additional sensors – each sensor simply adds another epipolar line to the overlay. The region contain in the intersection of all of the error bounds should contain an object, if it was in view of all of the sensors. Something to note is that more sensors beyond three do not necessarily reduce the search space further. In fact, if the error is not properly quantified, additional sensors may over constrain the problem and make the correct association impossible.

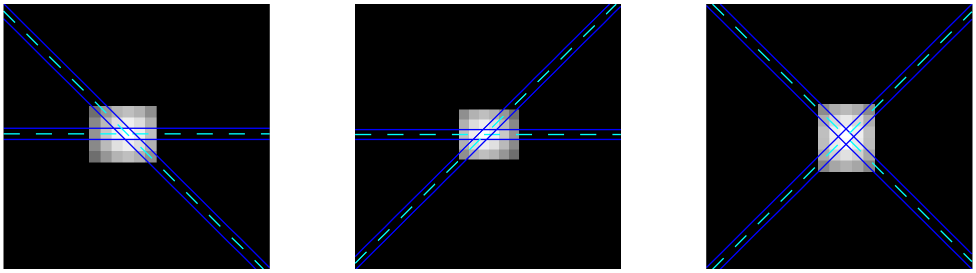

Figure 5.14: Epipolar overlay for three cameras with error bounds.

Even when an object falls in the intersection of epipolar lines in an ideal case like that in Figure 5.14, there is still a possibility that the there are actually multiple objects that aligned in this way by chance. We can work to estimate the probability of this happening based on the density of the stellar background. If we know the number of stars that are expected to appear in the image frame and these stars are equally likely to be anywhere in the image, then the probability that a star occurs in a single pixel is

$$P_s = \frac{\eta_s}{r_x r_y} \tag{5.59}$$

where r_x and r_y denote the resolution of the image along the x and y axes, respectively and η_s is the number of stars expected in the image. We can write the probability of a false association in terms of P_s and the area between the epipolar error bounds, A_i, as

$$P_f^{[i]} = P_s^{[i]} A_i \tag{5.60}$$

This is the probability that an object detected in one sensor is *not* the same as the object that appeared in the intersection epipolar lines. We assume that the likelihood of another non-star object appearing in the image and serendipitously aligning in a way that produces a false positive is negligible and therefore the only way for a false association to occur is from star clutter. So, to precisely characterize the probability of a false association we must estimate the number of stars that we would expect to fall in an image. We can achieve this goal through consideration of star catalog data [98, 6, 99]. First, let us consider the density of stars on the celestial sphere. This derivation will follow a similar analysis to that of the false positive rate for star identification with the pyramid algorithm [16]. Treating the celestial sphere as if were of unit radius, the stellar density, ρ_s, is

$$\rho_s\left(m_v\right) = \frac{N\left(m_v\right)}{4\pi} \tag{5.61}$$

where m_v denotes the limiting magnitude for detection of a star in a particular sensor and $N\left(m_v\right)$ is the number of stars brighter than that magnitude. So, the stellar density requires an estimate of the number of stars expected for a particular limiting magnitude. In [7], the author estimates $N\left(m_v\right)$ to be

$$N\left(m_v\right) \approx 6.57 e^{1.08 m_v} \tag{5.62}$$

This result is an estimate from the PPM star catalog of 378,910 stars [99]. In Figure 5.15 we see that this estimate agrees well with the Tycho-2 star catalog as well. In this figure, the dotted line indicates the number of expected stars for each visible magnitude from the equation above and the solid line is the true count from the 2.5 million star catalog [6]. The two plots deviate at the beginning and end, where the Tycho-2 catalog has limited coverage.

The area of the celestial sphere covered by a circular field of view with extent θ_{FOV} is [7, 16]

$$A_s = 2\pi\left(1 - \cos\left(\theta_{\text{FOV}}\right)\right) \tag{5.63}$$

So, the number of expected stars is equal to the area of the celestial sphere that a particular sensor covers multiplied by the stellar density.

$$\eta_s = A_s \rho_s \tag{5.64}$$

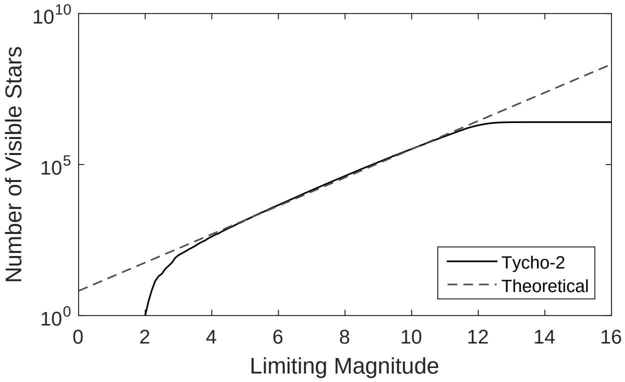

Figure 5.15: Number of visible stars vs. limiting magnitude [6, 7].

From Equations 5.62 and 5.63, the equation above becomes

$$\eta_s \approx 3.29\left(1-\cos\left(\theta_{\text{FOV}}\right)\right)e^{1.08m_v} \tag{5.65}$$

The number of stars expected an image is a function of the angular field of view of the sensor, θ_{FOV}, and the detection threshold, m_v, of the sensor. Figure 5.16 visualizes the number of expected stars for various field of view sizes and limiting magnitudes in the Tycho-2 star catalog [6]. For low limiting magnitudes (< 5), even wider field of view sensors are unlikely to see any stars. As the field of view and limiting magnitude increases, the number of stars that are likely to be viewed by the sensor increases rapidly.

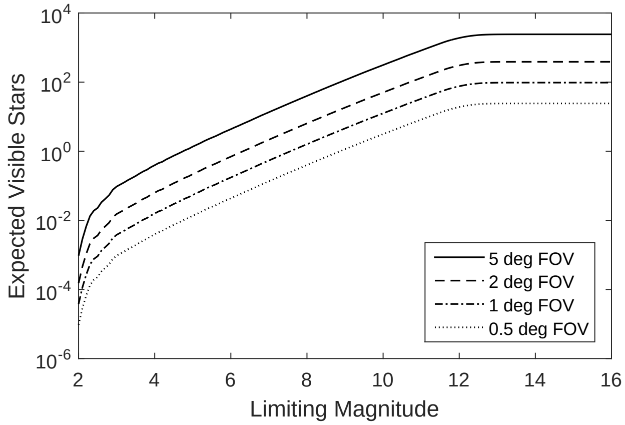

Figure 5.16: Expected number of stars vs. limiting magnitude in the Tycho-2 catalog.

Assuming that there is an equal probability for a star to be anywhere on the focal plane, the probability of a star falling in any square pixel area is approximately

$$P_s \approx \frac{3.29}{r_x r_y}\left(1-\cos\left(\theta_{\text{FOV}}\right)\right)e^{1.08m_v} \tag{5.66}$$

Of course, this is only an estimate since it assumes a homogeneous distribution of stars across the celestial sphere. In practice, it would be better to consider a worst-case scenario where the sensor is staring directly into the galactic disk. That way we would find a conservative estimate for the probability of a star appearing in any pixel. In other regions of the sky P_s would overestimate the probability of a star landing in a pixel, but it would never underestimate it.

Once we have a confident object association between sensors it is then possible to immediately compute the range of the object from any of the sensors involved. Define the angles between the measured line of sight vectors from two sensors, $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, and the relative position between the sensors, $\boldsymbol{r}_{1,2}$, to be

$$\begin{aligned} \theta_1 &= \cos^{-1}\left(\boldsymbol{u}_1 \cdot \boldsymbol{r}_{1,2}\right) \\ \theta_2 &= \cos^{-1}\left(\boldsymbol{u}_2 \cdot \boldsymbol{r}_{2,1}\right) \end{aligned} \tag{5.67}$$

It is from these angles that we are able to calculate the range of the object. The distance to the object along the line of sight vector $\boldsymbol{u}_1$ to from the first sensor is

$$D = \frac{\|\boldsymbol{r}_{1,2}\| \sin\theta_1}{\sin\left(\pi - \theta_1 - \theta_2\right)} = \frac{\|\boldsymbol{r}_{1,2}\| \sin\theta_1}{\sin\left(\theta_1 + \theta_2\right)} \tag{5.68}$$

So, the position of the object of interest defined relative to the first observer is

$$\boldsymbol{r}_{\text{rso}} = D\boldsymbol{u}_1 + \boldsymbol{r}_o \tag{5.69}$$

where $\boldsymbol{r}_o$ is the vector describing the position of the reference sensor. This range estimate has the additional benefit of allowing for discrimination between stars and non-stars in an image. Stars can be treated as existing at an infinite distance from the observer, so a range gating process will remove stars and only leave objects which are within a desired distance of Earth. An important consideration when computing the range of an object is that the baseline must be sufficiently large to allow for precise estimates of the range [19]. From the equation for distance above we see that as the baseline shrinks and the range grows, the angles θ_1 and θ_2 approach $\frac{\pi}{2}$ and the distance equation becomes singular. For baselines that are small relative to the range of the object small measurement errors are amplified and the resulting distance measure may not be accurate.

Another interesting consequence of a multi-sensor approach is that simultaneous tasking of multiple, widely separated observers allows us to effectively monitor regions of space. In some scenarios, the intersection of the field of view of multiple sensors encapsulates a closed volume. Figure 5.17 contains one such scenario. Through careful choice of the placement and pointing of individual sensors, then, it is possible to continuously observe regions of space. Any object that is measured with the multi-view algorithm discussed in this section is guaranteed to be in that volume of space. So, for example, if we wish to observe objects in a specific medium-Earth orbit (MEO), we might point the sensor system at a region that these objects are expected to pass through. Constructing one of these sensor systems to observe a region above Earth's pole could also provide continuous monitoring of a populated region of space.

In the simulated scenario in Figure 5.17, three sensors indicated by red markers are oriented according to the black arrows originating from each marker. Two sensors lie along the x axis separated by 2 units at $(-1, 0, 0)$ and $(1, 0, 0)$ and a third is located at $(0, 1, 0)$. All three of the sensors point at $(0, 0.5, 1)$. Starting from a grid of points in the three dimensional space, this simulation removed any points that were not in the field of view of all three sensors. The convex hull in the figure contains the remaining points and visualizes the intersection volume of the fields of view of the three sensors.

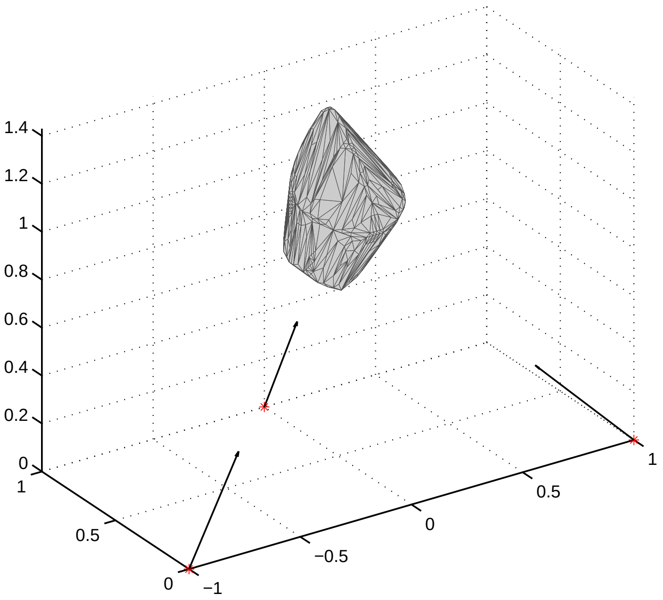

Figure 5.17: Numerically generated intersection of the field of view of multiple sensors.

5.4 Summary

This chapter has explored a geometric approach to data reduction for optical observers. Based on the derivations in Sections 3.3 and 3.4, we have shown several benefits to analyzing data in this way. Specifically, in Section 5.1, we showed that it is possible to reverse the effect of rotation about the boresight during an exposure. This correction leads to more precise measurements of objects contained in the image. Further, other types of traditional analysis are possible in the corrected space, possibly enabling a wide variety of alternative data processing techniques. In particular, Section 5.2 showed that it is possible to formulate the star subtraction routine originally developed in Section 4.3 in this corrected space. Through this type of analysis it is possible to remove stars with a common template even when there is extreme rotation around the boresight and the geometry of each star is dependent on its location in the image.

It is important to note that the results in Sections 5.1 and 5.2 are dependent on two key assumptions. The first is that the sensor has a small field of view. Choosing the appropriate field of view is dependent on the acceptable level of error and may be inferred from the neglected terms of Equations 3.22 and 3.25. The second constraint on these results is that, if the sensor is rotating, the axis of rotation must be constant. This constraint is slightly relaxed from the assumption of constant angular rate for the dynamic analysis in Sections 3.1 and 3.2, but applies to a similar set of scenarios.

Section 5.3 explored a different type of geometric image analysis. This work uses the derivations of the epipolar geometry for multi-sensor systems from Section 3.4 to augment traditional space object classification routines. We have shown that, with two or more sufficiently separated observers, it is possible to classify an object as a star or non-star without it exhibiting any apparent motion. Additionally, after a conclusive object classification the full position of the object is immediately available.

These multi-sensor algorithms have two key constraints placed on them. With the particular implementation derived here the sensors must adhere to the pinhole camera model. It is possible, however, to derive a model for the epipolar geometry among the sensors that uses higher fidelity models. The sensors must also be widely separated to enable the classification and triangulation procedures. In simulated scenarios, a baseline separation of 10 times the distance to the object from the observers was ideal.

These techniques offer a fundamentally new approach to looking at optical sensor data. By analyzing the geometry of sensor systems we gain a deeper insight into the nature of the sensor. This insight further enables a wider range of data processing routines that are applicable to a diverse set of optical sensors.

Chapter 6

Summary of Contributions

The work in this dissertation is intended to provide a set of general algorithms for handling optical data from diverse sources. As the number of space objects in Earth orbit grows, more sensors are necessary to maintain a precise space object catalog. The work here aligns well with the data-centric approach of DARPA's OrbitOutlook program which aims to extend data collection efforts to non-traditional sources such as commercial, university, and hobbyist level sensors [12]. With this broadening data collection effort comes a new set of challenges; low cost, non-traditional sensors do not necessarily provide the same precision of tasking and quality of data as the sensors that make up the Space Surveillance Network currently. So, in order to develop a centralized data processing pipeline, this work approaches data analysis from a fundamental standpoint. Only by leveraging the basic physics and geometry underlying optical systems will it be possible to fully utilize a diverse pool of sensors.

In all, this dissertation outlined six high-level approaches to image analysis enabled by the low-level sensor analysis derived in Chapter 3. Section 4.1 contained two algorithms pertaining to space object tracking and classification in optical imagery. These algorithms provide a novel approach to reducing optical image data based on the motion models from Chapter 3. Though we only considered two motion models in particular, a similar architecture may be built on any of the models depending on which set of assumptions best fit the system at hand. This type of general tracking and classification process is a powerful tool for developing a generalized pipeline that is capable of reducing data from a wide variety of sensors. For specific formulations of this pipeline, see [68] and [100].

The advanced shift and add process developed in Section 4.2 provides a higher-fidelity approach to the shift and add process that is typically applied for detection of dim space objects [15, 92]. Rather than simply assuming that objects of interest have a constant velocity across the image plane, we can relax this assumption and broaden the range of scenarios in which shift and add is beneficial. The results in this dissertation open up the shift and add process to a much broader range of sensors and space objects which may violate the typical assumptions. Maximizing the gain of low-quality sensors is a key aspect of utilizing a set of diverse observers to its full potential.

Star removal with the algorithm developed in Section 4.3 augments many of the other techniques discussed and developed here. Star clutter is often a problem for analysis of optical imagery because space objects of interest move independently of the stellar background. The presence of star clutter is perhaps most problematic when performing shift-and-add, but tracking algorithms can benefit from this process to a lesser extent. A convex formulation for the star removal problem is a novel, efficient method for removing stars. Though it does not have the fidelity of some existing approaches, the global process is flexible and may be extended to leverage deeper knowledge of an existing sensor. This work was presented at the 26th AAS/AIAA Space Flight Mechanics meeting [101].

The geometric approach to image analysis enables a way to account for non-ideal sensor tasking that includes rotation around the boresight. From Section 3.3, we know that the path of a star across the image plane of a sensor rotating with a fixed axis of rotation is a conic section. The results in Section 5.1 leverage this result to counteract the effect of rotation around the boresight. This rotation is undesireable because it leads to curvature in the background star signals and by extension reduces the accuracy of attitude knowledge. By performing image analysis with the motion profile of stars in mind, we are able to remove this effect and return to the level of accuracy that is typical of measurements of star signals without curvature. For more on this topic, see [102].

Finally, Section 5.3 extended our analysis to multi-sensor configurations. We have shown that it is possible to leverage the epipolar geometry of multiple sensors to not only triangulate space objects but also enhance the space object discrimination capability of an individual sensor. The results in Section 5.3 show that it is possible to build a probabilistic argument that a particular object in view of multiple sensors is not a star even when there is no apparent motion to differentiate it. In a centralized data processing pipeline, this approach, in conjunction with all of the other techniques described here, could provide higher-level analysis to intelligently merge data streams from individual sensors. This type of multi-sensor analysis has the potential to maximize extraction of information from individual sources to enhance overarching orbit determination routines.

Chapter 7

Bibliography

[1] T. S. Kelso. Analysis of the 2007 chinese asat test and the impact of its debris on the space environment. In *8th Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, HI*, 2007.

[2] T. S. Kelso et al. Analysis of the iridium 33 cosmos 2251 collision. 2009.

[3] Mike's Astrophotography Gallery & Blog. Comet Pan-STARRS Parting Shots, April 2013.

[4] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer vision, graphics, and image processing*, 116:87–116, 1988.

[5] B. Sease, B. Flewelling, and Y. Xu. Catalog-free angular rate estimation and online detecton of resident space objects. In *24th AAS/AIAA Space Flight Mechanics Meeting*, January 2014.

[6] E. Høg et al. The Tycho-2 catalogue of the 2.5 million brightest stars. *Astronomy and Astrophysics*, 355:L27–L30, March 2000.

[7] C. C. Liebe. Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):587–599, apr 2002.

[8] H. Shaub and J. Junkins. *Analytical Mechanics of Space Systems*. AIAA, 2nd edition, 2009.

[9] ExoAnalytic Solutions. SL 12 RB splits SATMEX 6 and MEXSAT 3.

[10] U. S. Air Force. Joint functional component command for space, 2012.

[11] H. Klinkrad. *Space debris: models and risk analysis*. Springer, Berlin; New York; Chichester, UK, 2004. 2nd print; 1 edition, 2006.

[12] T. Blake, M. Sanchez, and M. Bolden. OrbitOutlook : Data-centric Competition Based Space Domain Awareness (SDA). In *30th Space Symposium, Tech Track*, 2014.

[13] K. Park and D. Mortari. Planet or moon image processing for spacecraft attitude estimation. *Advances in the Astronautical Sciences*, 124 I(2):145–162, 2006.

[14] J. Christian. Optical Navigation Using Planets Centroid and Apparent Diameter in Image. *Journal of Guidance, Control, and Dynamics*, 38(2):192–204, 2015.

[15] T. Yanagisawa. The Stacking Method: the Technique to Detect Small Size of GEO Debris. In *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, 2003.

[16] D. Mortari, M. Samaan, C. Bruccoleri, and J. Junkins. The pyramid star identification technique. *Navigation-Alexandria*, pages 1–39, 2004.

[17] B. Spratling and D. Mortari. A Survey on Star Identification Algorithms. *Algorithms*, 2(1):93–107, January 2009.

[18] D. Lang, D. Hogg, K. Mierle, M. Blanton, and S. Roweis. Astrometry.Net: Blind Astrometric Calibration of Arbitrary Astronomical Images. *The Astronomical Journal*, 139(5):1782–1800, May 2010.

[19] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.

[20] B. Obama. *National space policy of the United States of America*. Executive Office of the President, 2010.

[21] T. S. Kelso. Celestrak. *Public Domain Satellite Tracking Data, URL: http://celestrak.com*, 2006.

[22] M. Wasson. Space situational awareness in the joint space operations center. Technical report, DTIC Document, 2011.

[23] Michael Morton and Timothy Roberts. Joint space operations center (jspoc) mission system (jms). Technical report, DTIC Document, 2011.

[24] M. Levesque. Automatic reacquisition of satellite positions by detecting their expected streaks in astronomical images. In *Proceedings of the Advanced Maui Optical and Space and Space Surveillance Technologies Conference*, 2009.

[25] K.. Henize, J. Stanley, C. O'Neill, and B. Nowakowski. Detection of orbital debris with geodss telescopes, 1993.

[26] B. Wallace et al. The Near Earth Object Surveillance Satellite (NEOSSat). In *Photonics North*, volume 5578, pages 1–7, 2004.

[27] T. Schildknecht, U. Hugentobler, and A. Verdun. Algorithms for ground based optical detection of space debris. *Advances in Space Research*, 16(11):47–50, 1995.

[28] R. L. Scott, B Wallace, and D. Bedard. Space-based observations of satellites from the most microsatellite. Technical report, DTIC Document, 2006.

[29] V. Kouprianov. Advanced Image Processing Techniques for Automatic Reduction of GEO Survey Data. In *8th US-Russian Space Surveillance Workshop*, 2010.

[30] Neil Goldstein et al. Ground testing of prototype hardware and processing algorithms for a Wide Area Space Surveillance System (WASSS). In *AMOS*, number ADA591374, 2013.

[31] C. Zhai et al. Detection of a Faint Fast-Moving Near-Earth Asteroid Using the Synthetic Tracking Technique. *The Astrophysical Journal*, 792(1):60, August 2014.

[32] P. Chu. Efficient detection of small moving objects. Technical report, DTIC Document, 1989.

[33] T. Schildknecht, R. Musci, and T. Flohrer. Properties of the high area-to-mass ratio space debris population at high altitudes. *Advances in Space Research*, 41(7):1039–1045, 2008.

[34] T. Kelecy and M. Jah. Analysis of high area-to-mass ratio (HAMR) GEO space object orbit determination and prediction performance: Initial strategies to recover and predict HAMR GEO trajectories with no a priori information. *Acta Astronautica*, 69(7-8):551–558, 2011.

[35] J. Crassidis and J. Junkins. *Optimal Estimation of Dynamic Systems.* CRC Press, 2nd edition, 2012.

[36] D. Vallado. *Fundamentals of Astrodynamics and Applications.* Microcosm Press, Hawthorne CA, 3rd edition, 2007.

[37] D. Shupe et al. The SIP Convention for Representing Distortion in FITS Image Headers. *Astronomical Data Analysis Software and Systems XIV ASP Conference Series*, pages 1–5, 2005.

[38] J. Morrison, B. Stamper, and D. Shupe. Correcting MIPS Spitzer Images for Distortion. *Astronomical Data Analysis Software and Systems*, 376:433–436, 2007.

[39] S. Howell. *Handbook of CCD Astronomy.* 2006.

[40] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *Astronomy and Astrophysics Supplement*, 117:393–404, 1996.

[41] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 2004.

[42] R. Gonzalez and R. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[43] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[44] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[45] P. Kovesi. Phase congruency detects corners and edges. *The australian pattern recognition society conference: DICTA 2003*, 2003.

[46] B. Sease, R. Koglin, and B. Flewelling. Long-integration star tracker image processing for combined attitude-attitude rate estimation. In *SPIE Defense, Security, and Sensing*, pages 87390Y–87390Y–12, May 2013.

[47] P. Gural, J. Larsen, and A. Gleason. Matched filter processing for asteroid detection. *The Astronomical Journal*, (1991):1951–1960, 2005.

[48] D. Mortari et al. CCD Data Processing Improvements. In *International Conference on Dynamics and Control of Systems and Structures in Space*, 2002.

[49] C. C. Liebe, K. Gromov, and D. Meller. Toward a stellar gyroscope for spacecraft attitude determination. *Journal of Guidance, Control, and Dynamics*, 27(1), 2004.

[50] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[51] M. Levesque. Evaluation of the accuracy of the dark frame subtraction method in CCD image processing. Technical report, DTIC Document, 2007.

[52] M. Levesque. Evaluation of the iterative method for image background removal in astronomical images. Technical report, DTIC Document, 2008.

[53] NASA Astronomy Picture of the Day. In the Glare of Alpha Centauri, June 2012.

[54] J. Anderson et al. Indication, from Pioneer 10/11, Galileo, and Ulysses Data, of an Apparent Anomalous, Weak, Long-Range Acceleration. *Physical Review Letters*, 81:2858, 1998.

[55] S. Turyshev, V. Toth, J. Ellis, and C. Markwardt. Support for temporally varying behavior of the Pioneer anomaly from the extended Pioneer 10 and 11 doppler data sets. *Physical Review Letters*, 107(8):1–5, 2011.

[56] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 46(12):1908–1913, 2001.

[57] D. Mortari, F. de Dilectis, and R. Zanetti. Position Estimation Using the Image Derivative. *Aerospace*, 2(3):435–460, 2015.

[58] D. Greenwood. *Principles of dynamics*. Prentice-Hall Englewood Cliffs, NJ, 1988.

[59] J. Schwartz. *Introduction to matrices and vectors*. Courier Corporation, 2001.

[60] K. Rangarajan and M. Shah. Establishing motion correspondence. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991.

[61] C. J. Veenman, M.J.T. Reinders, and E. Backer. Motion tracking as a constrained optimization problem, 2003.

[62] K. Davidson and A. Donsig. *Real analysis and applications: theory in practice*. Springer Science & Business Media, 2009.

[63] Y. C. Shiu and S. Ahmad. 3D location of circular and spherical features by monocular model-based vision. *Conference Proceedings., IEEE International Conference on Systems, Man and Cybernetics*, pages 576–581, 1989.

[64] G. Salmon. *A treatise on conic sections: containing an account of some of the most important modern algebraic and geometric methods*. Longmans, Green, 1900.

[65] J. Henderson et al. Intelligent sensor tasking for space collision mitigation. In *SPIE Defense, Security, and Sensing*, pages 76910L–76910L, 2010.

[66] R. Danescu, F. Oniga, V. Turcu, and O. Cristea. Long baseline stereovision for automatic detection and ranging of moving objects in the night sky. *Sensors (Switzerland)*, 12(10):12940–12963, 2012.

[67] R. Danescu, A. Ciurte, and V. Turcu. A low cost automatic detection and ranging system for space surveillance in the medium earth orbit region and beyond. *Sensors (Switzerland)*, 14(2):2703–2731, 2014.

[68] B. Sease and B. Flewelling. GEODETICA: A general software platform for processing continuous space-based imagery. In *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015.

[69] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, nov 2004.

[70] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, jun 2008.

[71] R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2:506–513, 2004.

[72] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[73] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[74] M. Primet and L. Moisan. Point tracking: an a-contrario approach. *MAP5 2012-06*, 2012.

[75] L. Bazzani, D. Bloisi, and V. Murino. A comparison of multi hypothesis kalman filter and particle filter for multi-target tracking. *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, 2009.

[76] G. Wahba. Problem 65-1: A least squares estimate of spacecraft attitude. *SIAM Review*, 7(3):409, July 1965.

[77] F. Markley and D. Mortari. How to estimate attitude from vector observations. *Advances in the Astronautical Sciences*, 103(PART III):1979–1996, 2000.

[78] L. Simms. Autonomous subpixel satellite track end point determination for space-based images. *Applied Optics*, 2011.

[79] D.D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and\nplanes with uncertainty models. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 696–702, 1998.

[80] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image feature points? *Electronics and Communications in Japan, Part III: Fundamental Electronic Science*, 86(1):1–10, 2003.

[81] R. Matt Steele and C. Jaynes. Feature Uncertainty Arising from Covariant Image Noise. In *CVPR'05*, volume I, pages 1063–1070, 2005.

[82] B. Zeisl et al. Estimation of Location Uncertainty for Scale Invariant Feature Points. *British Machine Vision Conference*, pages 1–12, 2009.

[83] J. Crassidis and J. Junkins. *Optimal Estimation of Dynamic Systems.* CRC Press, 2nd edition, 2012.

[84] S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, (January), 2004.

[85] P. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

[86] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, 1975.

[87] E. Kaufman, T. Lovell, and T. Lee. Optimal joint probabilistic data association filter avoiding coalescence in close proximity. In *Control Conference (ECC), 2014 European*, pages 2709–2714. IEEE, 2014.

[88] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous univariate distributions*, volume 2. Wiley & Sons, 1995.

[89] A Bhattachayya. On a measure of divergence between two statistical population defined by their population distributions. *Bulletin Calcutta Mathematical Society*, 35:99–109, 1943.

[90] G. Arce. *Nonlinear signal processing: a statistical approach.* John Wiley & Sons, 2005.

[91] C. León Vega, A. Elizálde Medrano, and E. Velazquez Lozada. Satellite system problems in mexico. In *Proceedings of the 56th Annual Meeting of the ISSS-2012, San Jose, CA, USA*, 2013.

[92] M. Shao, B. Nemati, C. Zhai, S. Turyshev, J. Sandhu, G. Hallinan, and L. Harding. Finding Very Small Near-Earth Asteroids Using Synthetic Tracking. *The Astrophysical Journal*, 782(1):1, February 2013.

[93] A. Heinze, S. Metchev, and J. Trollo. Digital Tracking Observations Can Discover Asteroids Ten Times Fainter than Conventional Searches. *The Astronomical Journal*, 150(4):125, 2015.

[94] S. Howell et al. The K2 Mission: Characterization and Early results. *Publications of the Astronomical Society of the Pacific*, 126(938), 2014.

[95] S. M. Beard, H. T. MacGillivray, and P. F. Thanisch. The Cosmos System for Crowded-Field Analysis of Digitized Photographic Plate Scans. *MNRAS*, 247:311–321, 1990.

[96] V. Kouprianov. Distinguishing features of CCD astrometry of faint GEO objects. *Advances in Space Research*, 41(7):1029–1038, jan 2008.

[97] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[98] M. A. C. Perryman et al. The HIPPARCOS Catalogue. *Astronomy and Astrophysics*, 323:L49–L52, July 1997.

[99] S. Röser and U. Bastian. The final PPM star catalogue for both hemispheres. *Bulletin d'Information du Centre de Donnees Stellaires*, 42(11):11–16, 1992.

[100] B. Sease, T. Murphy, B. Flewelling, M. Holzinger, and J. Black. Enabling direct feedback between initial orbit determination and sensor data processing for detection and tracking of space objects. In *SPIE Defense+ Security*, pages 94690M–94690M. International Society for Optics and Photonics, 2015.

[101] B. Sease, B. Flewelling, and J. Black. A class of convex optimization problems for template-based star subtraction. In *26th AAS/AIAA Space Flight Mechanics Meeting*, 2016.

[102] B. Sease and B. Flewelling. Polar and spherical image transformations for star localization and RSO discrimination. In *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015.