9-2022

Orbit Determination with Event-Based Cameras to Improve Space Domain Awareness

Conor M. Wisentaner

Orbit Determination with Event-Based Cameras to Improve Space Domain Awareness

THESIS

Conor Martin Wisentaner, Second Lieutenant, USAF

AFIT-ENY-MAS-22-S-148

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-MAS-22-S-148

ORBIT DETERMINATION WITH EVENT-BASED CAMERAS TO IMPROVE SPACE DOMAIN AWARENESS

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Space Systems

Conor Martin Wisentaner, B.S.

Second Lieutenant, USAF

September 15, 2022

AFIT-ENY-MAS-22-S-148

ORBIT DETERMINATION WITH EVENT-BASED CAMERAS TO IMPROVE SPACE DOMAIN AWARENESS

THESIS

Conor Martin Wisentaner, B.S.
Second Lieutenant, USAF

Committee Membership:

Lt Col Bryan D. Little, Ph.D
Chair

Richard G. Cobb, Ph.D
Member

Andrew S. Keys, Ph.D
Member

Abstract

The objective of this research is to assess the utility of a Commercial Off-The-Shelf (COTS) Event-Based Camera (EBC) for Space Domain Awareness (SDA) applications by evaluating its ability to produce data for orbit updates of resident space objects. Unlike traditional frame-based imaging sensors, the pixels on an EBC activate independently when a change in brightness is detected to produce a continuous data flow on a per pixel basis. This unique functionality provides much higher temporal resolution than traditional frame-based sensors, such that an EBC can generate far more data points from a single observation than a frame-based sensor. However, current COTS EBCs have less spatial resolution than current COTS frame-based sensors, and no research has yet investigated whether the increased volume of data from an EBC can compensate for the lack of spatial resolution of each data point.

Using a beamsplitter to provide equal data to an EBC and a frame-based sensor for observations of multiple RSOs, this research found that the volume of data produced by an EBC can compensate for the EBC's reduced spatial resolution to generate orbit updates of comparable accuracy to those produced by data from a frame-based sensor. This is especially true for single pass orbit updates, where the EBC provided a more accurate update than the frame-based sensor in 13 out of 14 cases.

I'd like to thank (in order of appearance) my first point of contact at AFIT while I was still an ambitious C1C hoping to write a thesis for a program that wasn't designed to include a thesis, Dr. Richard Cobb; my advisor who was willing to work with me to make that happen, Lt Col Bryan Little; and my two research companions who didn't mind giving up a few late nights and early mornings to help me conduct my research, Lt Col Dan Moomey and Capt Molly Wakeling. I think time is the most valuable thing we have, so I'm incredibly grateful for the time these four amazing individuals were willing to give me with an unclear return on investment.

I'd also like to thank the Cannery Crew and the general LT population at AFIT for providing a constant supply of dinner, good conversation, and occasional homework help to support me through these fourteen or so months.

I should also probably thank random chance for landing me here as well... a large number of the events that helped me get to where I am today were the result of random processes (like Mr. Fairman assigning me to Summer Space when my first preference was clearly Jump...) so I can't help but wonder where I'd be instead if some of those events didn't turn out as they did.

Table of Contents

Page

List of Figures

List of Tables

ORBIT DETERMINATION WITH EVENT-BASED CAMERAS TO IMPROVE SPACE DOMAIN AWARENESS

I. Introduction

1.1 Motivation

It is no secret that space is becoming increasingly congested [1]. For the past six decades, humans have been launching an ever increasing number of satellites into orbit, yet many satellites remain in orbit past their mission lifetimes and continue to pose a danger to active satellites as they cannot maneuver to avoid conjunctions. Collisions from this sort of conjunction have already occurred, for example, in 2009 a defunct Cosmos spacecraft collided with an active Iridium satellite at a relative velocity in excess of 11 $\frac{km}{s}$ [2], producing innumerable small pieces of debris that can in turn threaten another Resident Space Object (RSO) and cause further collisions [3]. The increasing rate of launches will only provide more opportunities for conjunctions to occur, and for each collision, space becomes less safe for the world. Short of removing the debris from space, the best way to help prevent future collisions like the Cosmos-Iridium collision is to accurately characterize the orbits of RSOs so that future conjunctions can be predicted and avoided.

At the same time, it is no secret that space is becoming increasingly contested. Modern militaries are rapidly increasing their dependence on space-based assets, such that the need to accurately track both red and blue spacecraft is key to any military operation in space. The identification, characterization, and understanding of space objects is of such importance to the United States Space Force (USSF) that General

John Raymond, Chief of Space Operations, chose to highlight Space Domain Awareness (SDA) as one of the USSF's five core competencies in the service's inaugural Space Capstone Publication [4].

Despite this, the rate at which objects are being launched into space is on track to exceed the ability of the USSF to maintain custody of all objects in the ever-growing RSO catalog. While the military, academia, and private industry have all done an excellent job of populating space, the problem of keeping space safe has largely been left to the USSF, whose sensors were originally intended to produce high quality orbit estimates of a relatively small number of satellites, rather than to provide orbit estimates for the entire catalog as it continues to grow. Because of this, there is an urgent need for a more diverse inventory of sensors that can contribute high quality observations to better characterize the orbits of RSOs [1].

Fortunately, the amateur astronomy community has taken an interest in the growing number of RSOs as well. As "trains" of recently launched Starlink satellites have become a common nighttime sight around the world, backyard hobbyists have become increasingly interested in tracking and observing satellites with commercial equipment, even using satellite observations as a hands on activity at astronomy outreach events because the movement of a satellite relative to the stars is easily visible to the unaided eye. This demand for commercial satellite tracking over the past decade has led to the development of several Commercial Off-The-Shelf (COTS) software packages that provide significantly more utility than anything available a decade ago, while also removing the need to spend time and effort to develop in-house RSO tracking software for general use cases.

Over the same decade, modern culture has embraced social media as a fixture of daily life, which has increased the demand for compact yet high resolution imaging sensors. The sensor development required to produce imaging sensors on flagship

smartphones [5] and digital cameras has benefited the hobbyist astronomy community as well, such that the sensors used on astronomy cameras now available to the public are far superior to those available a decade ago.

This research is intended to take advantage of both of these recent trends to explore a possible solution to the need for more and more accurate SDA sensors by investigating the practicality of using a COTS Event-Based Camera (EBC) setup to accurately determine the orbits of objects in space, with a specific focus on comparing the orbit updates produced by relatively new EBC technology to those produced by traditional frame-based technology.

An event-based camera, as the name implies, is a camera that records data as singular events rather than as frames [6]. This is fundamentally different from a traditional frame-based camera, which records data from each pixel over a defined time span to create an image frame. In contrast, an EBC records data on a per pixel basis when a pixel detects a change in brightness. Instead of producing discrete image files, an EBC produces a continuous data flow that consists of the time of each event, the polarity of the change in brightness, and the location in the sensor's pixel space at which the event occurred.

The concept behind EBCs, which are also referred to as neuromorphic cameras or silicon retinas, is to better mimic how organisms perceive sensory data on a subjective basis rather than on an objective basis. When you walk from an air conditioned building to a building with broken air conditioning, you perceive the air to be warmer, but your body does not give you an objective measure of the temperature. When you walk past a loud fan being used to circulate air in the warm building, you will eventually ignore the sound it generates as it fades into the background noise you become accustomed to. event-based cameras seek to emulate this subjective sensory reception, where the new and interesting elicits a response but the background can

be ignored[1].

This unique mode of operation makes EBCs theoretically well suited to be used for SDA applications. The ability to selectively report data for a bright object moving through a relatively dark field of view, with a timestamp unique to each event the object generates, allows for a potential advantage over traditional frame-based sensors because of the increased volume of data produced and the increased temporal resolution of each data point relative to the data produced by a frame-based sensor. Previous work has demonstrated that EBCs can detect stars and RSOs both at night and during the day [7, 8], but no published research has yet attempted to update the orbit of an RSO with data from an EBC and compare the update's accuracy to an orbit update made with data from a frame-based sensor receiving the exact same light input. This thesis aims to change that.

[1]Ironically, though, our eyes are the closest we have to an objective sense of the world around us. For as long as you stare at this sentence it will continue to be perceived as black text on a white page, for this reason the terms neuromorphic camera and silicon retina will not be used in this paper.

1.2 Research Objectives

The objective of this research is to assess the utility of current COTS EBCs for SDA applications from one of three possible levels of capability:

- Current COTS EBCs can be used for SDA applications to produce results comparable to or more accurate than contemporary COTS frame-based sensors.
- There is a valid use case for EBCs as SDA sensors, but the state of COTS options limits the ability of EBCs to produce results comparable to frame-based sensors.
- There is a fundamental limitation to EBC technology that prevents EBCs from being useful for SDA applications regardless of advances in technology.

This qualitative assessment will be informed by answering the following questions based on an analysis of orbit updates produced by an EBC observing the same satellites as a traditional frame-based sensor:

- How does the accuracy of orbit updates from EBC data compare to those generated from frame-based sensor data?
  - Are the errors of the orbit updates generated by each sensor comparable to each other, on the same magnitude as each other, or do they differ by orders of magnitude?
  - To what degree can the temporal resolution of each data point generated by an EBC compensate for the reduced spatial resolution of an EBC when compared to a frame-based sensor?
  - To what degree can the number of data points generated by an EBC compensate for its reduced spatial resolution?

- How does the current state of COTS EBC technology affect its utility for SDA applications?
  - Is EBC technology too noisy and immature [9] to be useful for SDA applications?
  - What, if anything, is limiting EBCs from being useful for SDA applications?
  - Can these limitations be overcome with further development of the technology, or are they due to fundamental aspects of EBC technology?

The final analysis of whether or not EBCs are suited for SDA use will be informed by the quantitative results of the experiment and the answers to these research questions.

1.2.1 Assumptions and Limitations

The fundamental assumption made in this research is that the EBC and the Complementary Metal-Oxide Semiconductor (CMOS) sensors being used to gather data are receiving the exact same light and that all major sources of error, such as the effects of atmospheric disturbances, optical aberrations, and other effects beyond the focal plane of the telescope will be identical at each sensor.

This is assumed because the sensors are mounted on a nominally 50:50 split beamsplitter attached to the back of a telescope, such that any light the telescope receives is halved before being transmitted or reflected to the CMOS and EBC sensors. The beam splitter is in fact not perfect[2], but for the purposes of this research it is assumed to be sufficient to provide a direct comparison between the EBC and the CMOS sensors.

[2]For the specific reflectance and transmission curves, see Figure 11 and Figure 12 in Section 3.1.1.2.

It was also assumed that the RSOs for which orbit updates were generated would not maneuver between successive Two-Line Element (TLE) epochs and were not spinning or otherwise producing a signature that would be more easily detected by one sensor than the other. Most of the RSOs bright enough to be observed were rocket bodies, and in the case of the ISS, data was only used to make general inferences about the abilities of the two sensors.

The fundamental limitation of this research is that it deals with the uncontrolled environment of the real world. Unlike a software simulation, there is a near endless list of both environmental and hardware factors that could influence the data collected by both sensors. Steps will be taken to reduce the impact of major sources of error, such as atmospheric disturbances and sensor noise, but not every source of error can be corrected for or even perceived.

A limitation resulting from this is that observations produced by one observer at one point on the Earth are limited in accuracy due to uncertainties inherent to real-world observations. While this will be described in more detail in Section 2.1.4, the result of this is that orbit updates will not be of comparable accuracy to official TLEs and can only be compared against each other.

Because not every error can be quantified, the analysis will rely on the assumption that all major sources of error will affect the sensors equivalently and not affect the comparison between the EBC and the CMOS. While it is possible that differential flexure between the sensors and the beam splitter could introduce some error as the weight of each sensor affects its exact alignment, it is assumed that such a small difference will not have a significant impact on the ultimate results of the analysis.

1.3 Document Overview

This document is organized as follows. Chapter II provides an overview of background information on the equipment used and previous relevant research. Chapter III describes the equipment setup and steps used to gather and process the data. Chapter IV describes the quality of the orbit updates provided by each sensor and analyzes the relative abilities of each sensor. Chapter V restates the evaluation criteria and discusses the suitability of current COTS EBCs for SDA applications.

II. Background and Literature Review

2.1 Technical Background

2.1.1 Frame-Based Camera Technology

Initially invented and used as scientific instruments in the 1960s and 1970s, Charged Coupled Device (CCD) sensor and Complementary Metal-Oxide Semiconductor (CMOS) sensor technology has continuously improved in the years since as demand for ever more capable imaging sensors has increased [5]. Now, these digital frame-based sensors are the sort of sensors that are present on everything from dedicated cameras to cars and smart doorbells as imaging sensors have become ubiquitous in modern life.

Though CCD sensors process received photons slightly differently than CMOS sensors, both operate on the basis of frames as the electrons produced by each picture element (pixel) are collected together after each exposure and sent together to an image processor.

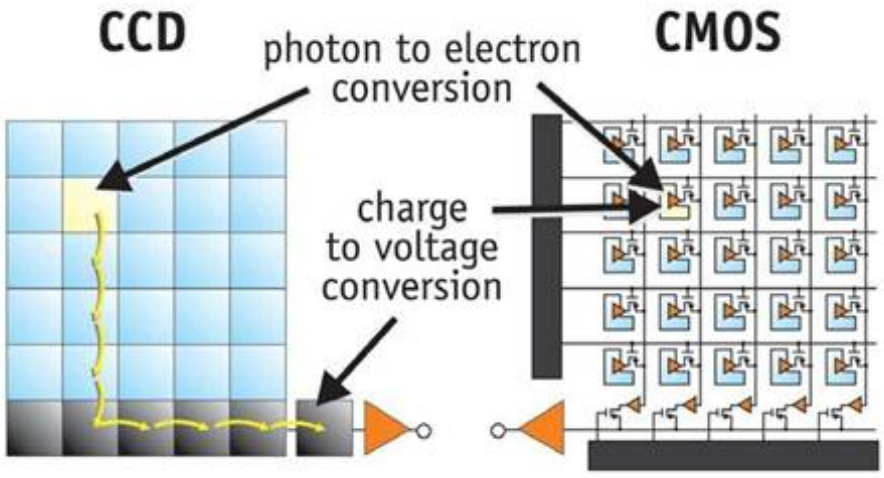


Figure 1: Comparison of CCD and CMOS sensor architecture [10].

While CMOS sensors do perform some data processing on a per pixel basis as

shown in Figure 1, they do not qualify as event-based sensors because each pixel records photons on an objective basis and the data from each pixel is reported together at the end of an exposure.

2.1.2 Event-Based Camera Technology

As alluded to in Section 1.1, the concept behind the development of the Event-Based Camera (EBC) was to offer a sensor that mimics the biological processes by which human senses perceive changes on a subjective basis, rather than on an objective basis [11]. While the original intent was to synthesize a functioning neuromorphic system in silicon, a result of this was that EBCs can detect changes in the sensor's Field of View (FOV) without producing data on the whole FOV that would prove computationally expensive for machine learning systems [6]. For example, an EBC could be used to highlight a bicyclist in the road, such that their movement would generate events, but without recording much extra data from the part of the background that is not changing.

Figure 2: Example of a cyclist being observed by a frame-based sensor and an EBC. While the frame-based sensor can see the cyclist, the EBC highlights their position much more clearly [6].

Despite many innovative use cases that take advantage of an EBC's unique mode of operation, the spatial resolution of EBCs remains relatively low. When compared

to modern CMOS sensors that have benefited from the drive to put a great camera on every smartphone [5], EBCs have relatively large pixels and small sensor sizes that provide limited spatial resolution. However, their ability to operate pixels independently of each other allows for far superior temporal resolution when compared to modern CMOS sensors, so they are theoretically well suited to tasks such as satellite detection that can make use of increased temporal resolution at the expense of spatial resolution.

The high temporal resolution of EBCs is possible because each pixel acts independently to produce an event as soon as a change in the brightness level is detected:

$$event = [x, y, p, t] \tag{1}$$

where the camera outputs the event's x position and y position in pixel space, the polarity of the event in terms of the change of brightness detected, and the time at which the event was observed[1]. Because events occur on a per pixel basis the temporal resolution of the sensor is limited at the pixel level rather than at the row or sensor level, which allows EBCs to produce data as a continuous flow rather than as discrete frames.

[1]This description of the data stream is simplified to convey the difference between EBCs and frame-based sensors. For a more complete description of the data flow, see Section 3.3.2.

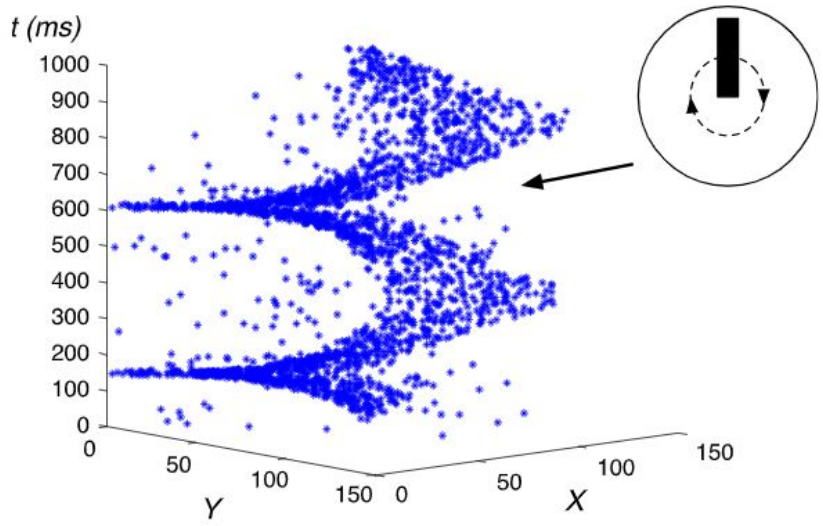


Figure 3: Example of an EBC recording a spinning bar [12].

This is demonstrated in Figure 3, where the continuous flow of data recorded by an EBC observing a rotating bar can be reconstructed into a three dimensional representation of the bar's motion. A frame-based video of the same bar would only show it at certain points throughout its rotation, and because the data would come in discrete frames, the only way to obtain data on the position of the bar between frames would be to interpolate. Extending this to the case of a satellite crossing the FOV of a Space Domain Awareness (SDA) sensor in a few seconds, the difference between a CMOS sensor and an EBC sensor could be the difference between a dozen data points and a few hundred data points. The integration time over which a frame-based sensor collects data on a satellite crossing its FOV often only allows for certain start and end points, such that each frame of a satellite streak can only contain two certain data points to be used for an orbit update. However, every point along a streak recorded by an EBC could be used as a data point to provide an orbit update for an object of interest[2]. This continuous data flow has the potential to provide a significant advantage over traditional cameras when applied to SDA because of how

[2]Within certain limits, which are discussed in Section 4.2.

many more data points an EBC can provide from just a single pass of an object.

Although data could be interpolated between the endpoints of a satellite streak in a frame from a CMOS sensor to create a few hundred data points, the fact that satellites travel through curved orbits similar to the star trails shown in Figure 5 would make a linear interpolation like this inaccurate for applications that require high precision data.

2.1.3 Astrometry and Plate Solutions

Astrometry is the identification and measurement of stars within an image, and a plate solution is an application of astrometry that uses the known locations of stars[3] in an image to determine the location of an image on the celestial sphere. Combined, they can be used to determine the position of unknown objects such as satellites within an image based on the locations of known objects in the image.

Astrometry is made possible by the fact that, on both the spatial and temporal scales with which this research is concerned, all stars can be considered to inhabit a "celestial sphere" surrounding the Earth with fixed positions on the sphere based on how they appear to an Earth-based observer [13]. Although this may sound like a step backwards in the understanding of the stars, it is a very useful simplifying assumption that provides the basis for optical satellite tracking.

An everyday application of this assumption is the use of constellations to describe the positions of stars. Anyone who has ever pointed out the North Star to a friend while stargazing by describing it as being roughly in line with the edge of the Big Dipper's scoop has implicitly assumed the stars occupy a celestial sphere, despite the fact that not even the stars in the big dipper are close to each other in space.

This example is how astrometry works, albeit with much higher fidelity. Rather

[3]Other celestial objects such as asterisms and distant galaxies can also be used for astrometry as well, but are less ideal due to not being point light sources.

then relative to each other, the positions of the stars on the celestial sphere are described with angular coordinates relative to the center of the Earth.

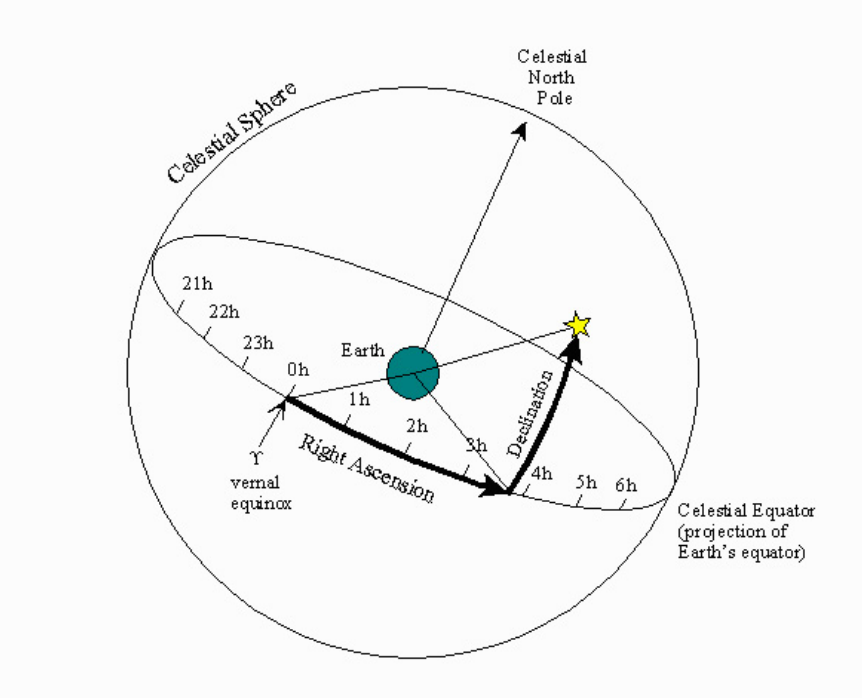


Figure 4: Depiction of the celestial sphere surrounding Earth with right ascension and declination axes labelled [14].

The two angles used in this research to describe a position on the celestial sphere are geocentric right ascension and declination. Topocentric right ascension and declination can also be used to describe stars because the radius of the Earth is insignificant on the scale of stellar distances, but this substitution is not valid for the satellites this research is concerned with because the radius of the Earth is not negligible when compared to the distance of even the furthest satellites.

Right ascension is measured in hours, minutes, and seconds of angle east to west around the plane of the Earth's equator, beginning and ending at the vernal equinox [15]. While right ascension can be converted to degrees for calculations, it is often stylized in the form of $HH^hMM^mSS.SSS^s$ to provide a more intuitive understanding of an object's location when the number is not being used in an automated calculation.

This is done because, as Figure 4 shows, a point on the celestial sphere with a right ascension of $01^h00^m00.000^s$ will be visible to an observer on Earth in the same spot in the sky every 24 hours as the Earth rotates about its axis over the course of a sidereal day[4].

Declination is measured in degrees, minutes, and seconds of angle above or below the plane of Earth's equator within the range of $\pm$ 90 degrees [15]. Because this is a geocentric measure, declination can be thought of as the degree to which an observer standing on the plane of the equator in the center of the Earth would need to look up or down to see a point on the celestial sphere.

The reason that these angles are geocentric is to provide an absolute basis from which to measure locations on the celestial sphere, and in particular to differentiate them from azimuth and altitude.

Figure 5: An example of stars streaking across the night sky relative to a fixed observer.

Because the Earth rotates with respect to the celestial sphere as shown in Figure 5,

[4]A sidereal day is actually 23:56:4.09, but 24 hours is a convenient simplification to provide a baseline understanding.

azimuth and elevation are only useful from one location for a specific point in time. Unlike right ascension and declination that provide constant locations on the celestial sphere, azimuth and elevation cannot be used alone to provide an objective measure of objects on the celestial sphere because they rely on a non-inertial observer rather than a fixed inertial one.

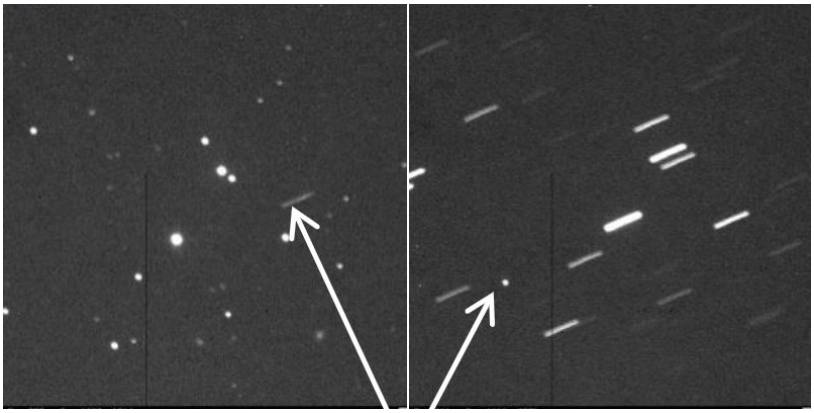


Figure 6: Comparison of sidereal tracking (left) to tracking at the rate of an object's motion (right) with the object's position noted, adapted from [16].

That said, the rotation of the celestial sphere with respect to a fixed observer can be taken advantage of to generate data to be used for astrometry. Because the celestial sphere rotates around a known axis at a known sidereal rate, certain telescope mounts can be aligned parallel to Earth's axis of rotation and set to rotate opposite the sidereal rate to track the movement of the celestial sphere relative to the telescope's fixed location. This method, known as sidereal tracking, allows a fixed observer to follow the celestial sphere's motion such that the position of stars remain constant points within a telescope's FOV as shown in Figure 6. By maintaining stars as point light sources, their relative positions within an image are preserved and can be used to generate a plate solution for the image.

Other tracking methods, such as tracking at the rate of the satellite's motion or not tracking at all, do not maintain the stars as point sources of light and cannot be used to generate plate solutions to be used for orbit updates.

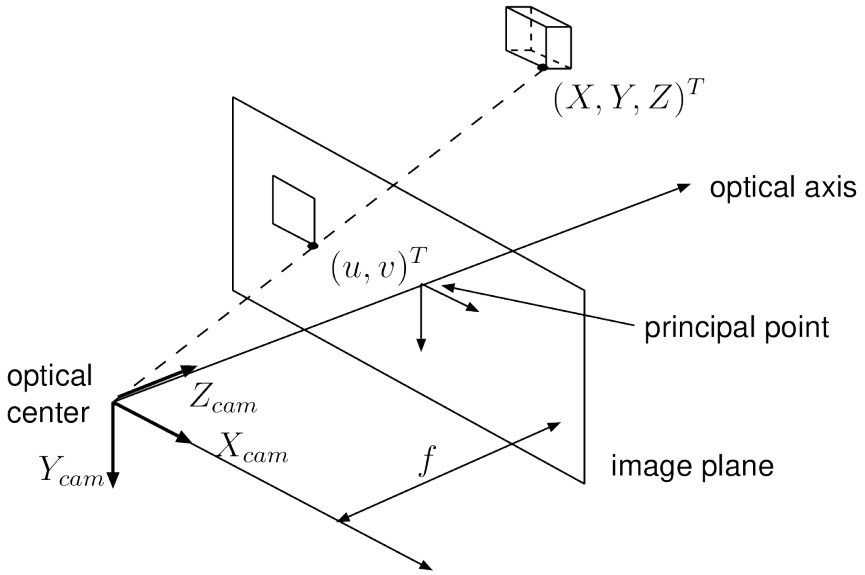


Figure 7: Model of a pinhole camera showing a three dimensional object projected onto a two dimensional image plane.

The process of plate solving an image of a star field takes advantage of the assumption of a celestial sphere by reverse engineering the model of the pinhole camera shown in Figure 7. An image is recorded as light from a distant three dimensional object projects onto a two dimensional focal plane, so, it is also possible to project the two dimensional plane of the recorded data back onto the three dimensional surface visible from the camera. Because the three dimensional surface in this case is the celestial sphere and the locations of major stars have been well characterized, pattern matching algorithms between the image and known star locations can be used to determine where on the celestial sphere the camera imaged when it recorded the star field [17]. Once matched with the surface area of the celestial sphere that it covers, an image is considered to have a plate solution, and the position of every pixel in the image can be converted to right ascension and declination based on their location relative to certain points in the image using the linear World Coordinate System (WCS) translation.

This process generally assumes that the two dimensions of the image sensor can be

mapped into the three dimensions of the celestial sphere while ignoring the distortions that this inevitably causes. However, this is only useful up until the point at which a plate solution is generated, at which point the patterns of stars in the image may not perfectly match the locations of known stars. Using these known patterns of stars in the sky, the plate solving process can characterize the offset between observed and expected positions to calculate a nonlinear Simple Imaging Polynomial (SIP) distortion correction to the image to be included with the plate solution data. So, despite the theory of plate solving an image making use of an initial simplifying assumption, the actual plate solution accounts for distortion and provides the data necessary to correctly translate from the pixel space of the sensor to geocentric right ascension and declination.

2.1.4 Orbit Determination

Unfortunately, despite the best efforts of the global engineering community, nobody has yet been able to convince a satellite to remain in a stable orbit that conforms to the sort of models found in introductory astrodynamics textbooks. Instead, to the dismay of undergraduate engineering students the world over, the orbit of every satellite is subject to many small effects known as perturbations that cause slight changes to the orbit over time such that even the highest-order models that account for the minute effects of solid Earth tides cannot correct for the uncertainty inherent to perturbed orbits and cannot reliably predict the position of a satellite for longer than a few weeks.

The solution to the problem of maintaining custody of satellites is to consistently observe them travelling through different sections of their orbits and use complex mathematical models to update their previous orbital states with new observation data. In doing so, incremental updates to the catalogued orbits of satellites can

be made to counter the incremental changes made to their orbits by perturbations. While this data collection can take many forms, the most intuitive method is to use a camera mounted behind a telescope to observe satellites passing through star fields and use the data from a plate solution to determine the position of the satellite at the time when the picture was taken. This process, which will be described in more detail in Chapter III, provides data on the relation of an unknown position of a satellite to the known position of stars within the frame. With enough data points, especially from observers with access to different portions of a satellite's orbit, a model can be used to determine the orbit that best fits the data points to provide an update to the orbit of the satellite.

This general method takes advantage of the fact that the orbit of a satellite can only be circular or in the form of an ellipse with one focus at the center of the Earth. With these constraints, each additional data point can be used to better narrow the scope of possible orbits that the satellite can occupy. So, to obtain the best possible update of an orbit, there should be as many data points as possible from as many geographically dispersed observers as possible.

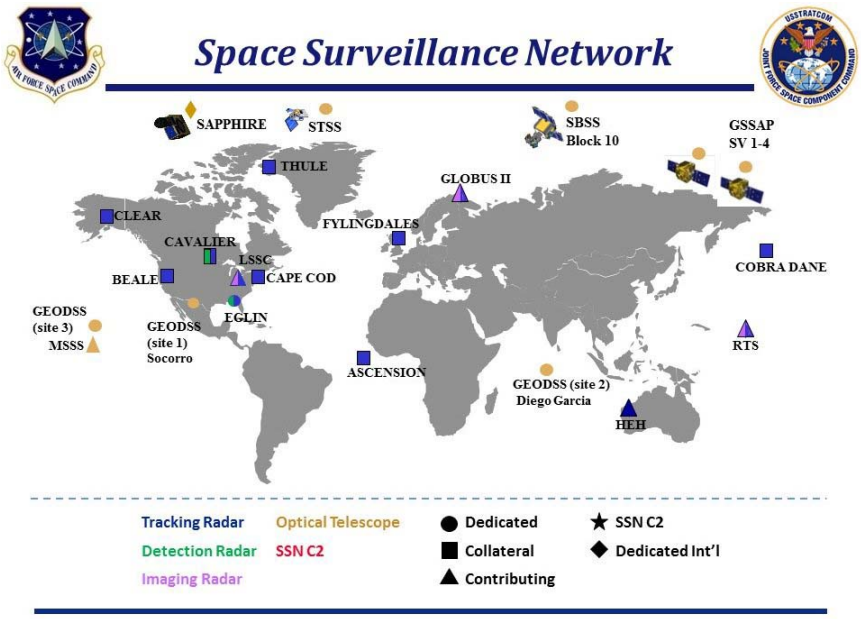


Figure 8: Map of the Space Surveillance Network of satellite observing stations used by the United States Space Force (USSF) to maintain custody of satellites [18].

This principle provides the impetus for the USSF's Space Surveillance Network (SSN) as shown in Figure 8, which is made up of a variety of sensor types spread around the globe to collect high quality data on the orbits of satellites.

The converse of this is that the ability of a single observer that can only measure the same section of an orbit will be limited in the quality of orbit update that it can produce.

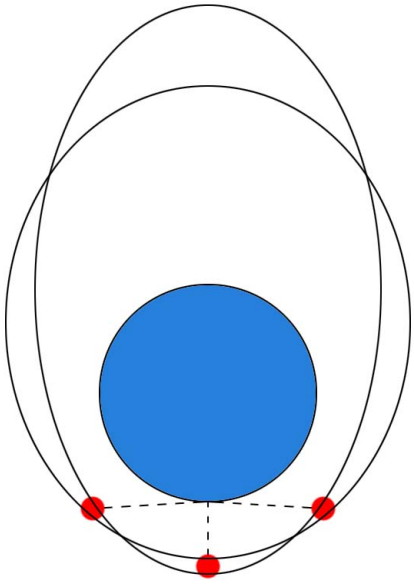


Figure 9: Depiction of two possible ellipses passing through the same uncertain measurements.

This limitation is demonstrated in Figure 9, which shows how the sky accessible to a single observer on one side of the Earth can only provide partial information on the orbit of a satellite. There is some degree of uncertainty inherent to Earth based observations due to rapidly varying atmospheric distortion, such that multiple orbits could be possible through each set of measurements from a single observer at a single location.

These fundamental limitations of a single observer at a single location will significantly impact the research to be done for this thesis. Unfortunately, building a global network of observers is prohibitively expensive even with the resources available to AFIT, so this research will focus on comparing the ability of two different sensors making observations through the same telescope at the same location to determine which sensor can produce more accurate orbit updates from the same light input. It is expected that these updates will not exceed the accuracy of the updates provided by a proper Two-Line Element (TLE) from the SSN, to do so would be beyond the

scope of this research.

2.2 Research Background

While no research has yet attempted to directly compare orbit updates produced by data from an EBC with those produced by a CMOS sensor, there is a wealth of information on the operations and limitations of EBC technology from both within and outside of the Air Force engineering community.

2.2.1 Event-Based Camera Characterization

In 2021, Captain Jessica Horn characterized the ability of an EBC to survive tests typically used to qualify satellite flight hardware to demonstrate that an EBC could potentially be used for on-orbit applications. Though her work primarily focused on the use case of EBCs in the environment of space, her finding that an EBC has a significantly smaller solar exclusion angle than traditional frame-based sensors [19] has implications for ground-based SDA as well. Ground-based optical sensors have a solar exclusion angle that is generally limited by the presence of the Sun anywhere in the local sky, so the potential for an EBC to detect objects near the Sun during the day would certainly fill an SDA gap for ground-based optical observations.

Capt Horn's work also highlighted other aspects of EBCs which can make them particularly well suited for SDA applications, such as their Size, Weight, and Power (SWaP) characteristics. Her work noted that because EBCs only output data when events occur, they draw less power than frame-based sensors, and because events are output as a continuous flow as they occur, EBCs experience less latency than frame-based sensors that have to wait for an exposure to finish before outputting data.

In 2020, Flight Lieutenant James Boettiger performed similar characterization

research on EBCs with a focus on object detection and tracking [9]. While his work focused on comparing the abilities of frame-based sensors and EBCs to develop target tracks for objects moving in a three dimensional space with respect to the sensor, his work on reducing the noise inherent to data from current Commercial Off-The-Shelf (COTS) EBC sensors could prove useful to this research.

2.2.2 Event-Based Cameras for Space Object Detection

While EBCs have existed in a functional form since the early 1990s [11], their applicability to observing objects in space was first noted in a paper for the Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) in 2017 written by Dr. Gregory Cohen and colleagues from the Western Sydney University [8]. The team of researchers noted that EBCs had successfully demonstrated potential in a number of other fields of research, and decided that the unique functionality of EBCs would likely yield promising results when applied to detecting objects in space as well.

For their experiment, they set up two EBCs and a CCD camera on the same mount and boresighted the telescopes togther so that, while each sensor had its own scope, the three sensors would all observe roughly the same portion of the sky. With this setup, they were able to determine that both stars and satellites could be observed by EBCs, including dimmer objects far away in Geosynchronous Equatorial Orbit (GEO). With the signal to noise ratio provided by the light gathering ability of their large telescopes, the group demonstrated the ability of EBCs to observe Low Earth Orbit (LEO) satellites during the day. While intended as a proof of concept rather than a proper test of the utility of EBCs for SDA applications, their research succeeded in proving the viability of EBCs for observing objects in space in a manner that is not constrained by the limitations of traditional frame-based sensors.

Their results also demonstrated that stars could be observed while the telescopes providing signal to the EBCs were tracking at a sidereal rate. Though their paper does not explicitly mention this as a finding, it provides a baseline for future work into generating plate solutions from EBC data. While it was mentioned in Section 2.1.3 that sidereal tracking maintains stars as stationary light sources within the FOV of a sensor, and in Section 2.1.2 that EBCs only generate events when they detect a change in brightness, it is not impossible for EBCs to detect stationary stars within their FOV as shown in Figure 6. Due to atmospheric scintillation, or more generally how stars "twinkle", stars within the FOV of the EBC will generate some events as their light is continuously disrupted by the atmosphere on its path to the sensor. The brightness threshold for stars to generate events due to scintillation is much higher than the threshold required for the star to be detected by a CMOS sensor, but the fact that they can be observed at all implies that future work may be able to generate plate solutions from EBC data only without the aid of a frame-based sensor.

Some of this future work came in the form of the same group's paper for AMOS in 2018, titled *Approaches for Astrometry using Event-Based Sensors* [20]. Using a very similar experimental setup to their 2017 paper, Dr. Cohen and his team demonstrated the ability to produce a plate solution from data provided by an EBC staring at a Resident Space Object (RSO) in GEO. Event data from 420 seconds of stars moving through the FOV of the sensor at the sidereal rate was compiled to generate a map of the stars that had been detected, which was then plate solved to generate a position solution from the EBC data.

While promising, this method of generating plate solutions with an EBC is not directly applicable to the process that was used for this research to generate plate solutions because it involved staring at a GEO object and letting stars streak by, rather than staring at a star and letting a LEO object streak by. More work will still

need to be done to generate plate solutions from data collected by an EBC tracking a star at the sidereal rate.

2.2.3 Event-Based Cameras for Space Domain Awareness

The bulk of the research concerning the use of EBCs for SDA applications has come from AFIT, with the first AFIT thesis that researched the use of EBCs for SDA applications being the work of Captain Joseph Bacon in the 21M year group. Though his experimental setup resembled that of Dr. Cohen's with the use of an EBC behind a large telescope, his research differed in that he explicitly sought to characterize the capability of EBCs for SDA applications [7], which would provide data to inform decisions on possible operational use of EBCs. Specifically, he sought to determine the look angles and angular rates of objects with EBC data, he attempted to generate orbit updates with EBC data, and he attempted to perform astrometry on the EBC data.

Unfortunately, due to some limitations of the hardware used for his research, he was unable to generate all of the intended results. The scope chosen to provide light to the EBC had a FOV with too few stars to provide a plate solution [7], and because of this, his ability to produce an orbit update was limited to manual calculations between the few stars in the EBC data and the satellite streaks without the WCS or SIP data that a plate solution would have provided. While he did produce two orbit updates, neither used a co-witness CMOS sensor to produce an accurate plate solution, and neither yielded results accurate to within a degree of the intended target. His problems were further compounded by the COVID-19 pandemic, which denied him many of the face to face interactions that are key to hardware based research projects like this. However, his work provided a proof of concept from which this research could expand, while taking into account the issues that prevented him from

achieving his research goals.

III. Methodology

Introduction

This chapter will discuss the equipment and methodology used to collect data and process data into orbit updates. Its depth is not intended for it to function as a complete checklist of all steps taken to generate the data, rather, it is intended to explain what decisions were made to optimize the quality of data collection and processing.

3.1 Data Collection

3.1.1 Instrumentation

The key to this research is the hardware setup that allowed both an Event-Based Camera (EBC) and a Complementary Metal-Oxide Semiconductor (CMOS) sensor to receive exactly the same light from the same telescope via a beam splitter, as detailed in Figure 10:

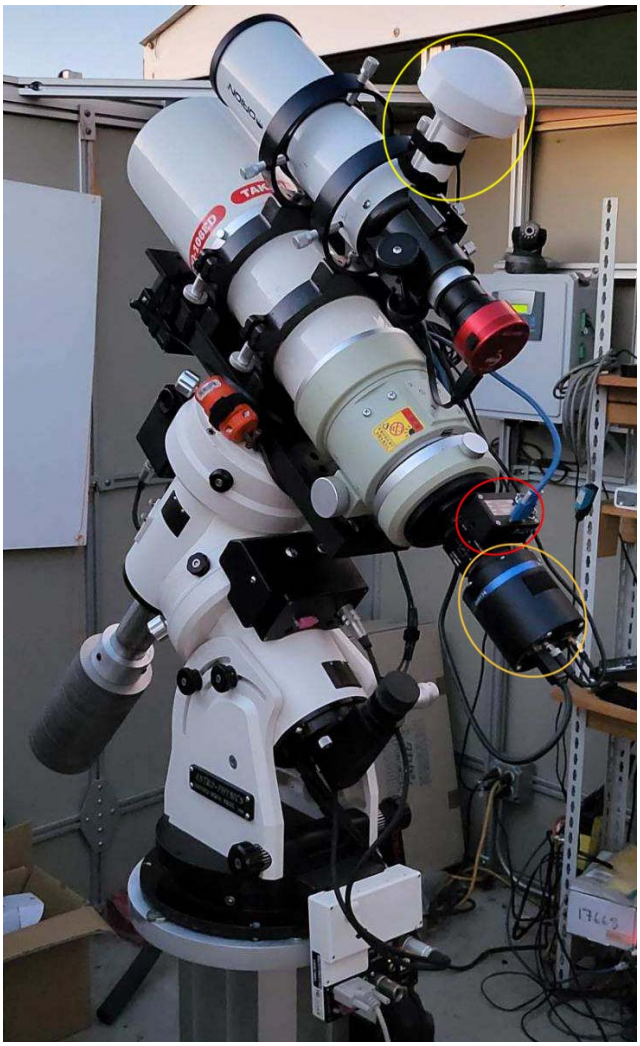


Figure 10: Image of instrumentation used to collect data for this research. The beamsplitter is directly attached to the telescope, with the cylindrical CMOS sensor (orange oval) extending directly backward from the telescope and the EBC (red oval) located on the other output path of the beamsplitter between the telescope and the CMOS sensor. The GPS receiver is shown inside the yellow oval.

The data was collected using the following instruments:

- Takahashi FSQ-106ED refractor telescope
- THORLABS CCM1-BS013 cage cube-mounted non-polarizing 50:50 beamsplitter, 400 - 700 nm
- QHY174M-GPS CMOS sensor

- iniVation DVXplorer EBC sensor

Other hardware, such as the telescope mount, focusing equipment, and assorted adapters also supported this research but will not be detailed because they did not directly impact the light data received by the two sensors.

3.1.1.1 Takahashi FSQ-106ED

The telescope used for this research was a Takahashi FSQ-106ED refractor telescope mounted on an AstroPhysics 1100 GTO mount.

Metric	**Value**
Focal Length	530 mm
Effective Aperture	106 mm
Resolving Power	1.09″
Limiting Magnitude	11.9 mag

Table 1: Takahashi FSQ-106ED Specifications [21]

The beamsplitter was mounted to the scope's T mount adapter and the light was focused using a FocusLynx computer controlled autofocuser. The proper distances to the focal planes of the two sensors could not be perfectly matched within a millimeter of each other, so a focus was chosen that left both unfocused by about the same offset in light path. While not ideal, no impact to research results was caused by this compromise.

3.1.1.2 THORLABS CCM1-BS013

The beamsplitter used for this research was a THORLABS CCM1-BS013 cage cube-mounted non-polarizing 50:50 beamsplitter designed to work with light of wave-

lengths from 400 nm to 700 nm.

Metric	Value
Reflected Beam Deviation	$90°\pm5'$
Transmitted Beam Deviation	$\leqslant 5'$

Table 2: THORLABS CCM1-BS013 Specifications [22]

A beam deviation of up to $5'$ could be detrimental to research that depends on arcseconds of accuracy. However, because an error introduced by the beamsplitter can only occur in the short distance between the beamsplitter and the CMOS or EBC, the actual effect of any error is minimal because the $5'$ will not cause significant deviation across this short distance. Assuming that the distance between the beamsplitter and the sensor is 1.5 cm and the error is at its maximum of $5'$:

$$\text{offset} = 1.5\text{cm} \times \tan(5') = 21.817\mu m \tag{2}$$

Which is about 3.7 pixels on the CMOS and 2.4 pixels on the EBC, though for light from distant objects that enters effectively perpendicular to the face of the beamsplitter, it is unlikely that the beam will deviate by the full $5'$ of the error tolerance.

Additionally, the experimental setup controls for any error introduced by the beam splitter by design because the location of stars within an image is only determined after light passes through the beamsplitter. If an image were to be distorted, that distortion would be corrected for in the data provided by the plate solution in the form of second order distortion correction coefficients contained in the Simple Imaging Polynomial (SIP) data of the Flexible Image Transport System (FITS) header [**?**]. While not ideal, this source of error cannot be avoided because the beamsplitter is

key to comparing the EBC to the CMOS on an equal basis.

The experimental design also corrects for differential flexure between the telescope and the sensors because the pixel space of the EBC is mapped to the pixel space of the CMOS. Even if there were to be an issue with differential flexure, its impact would be mitigated because the location of data points in the EBC pixel space is translated back to the pixel space of the CMOS before any calculations of right ascension or declination are made.

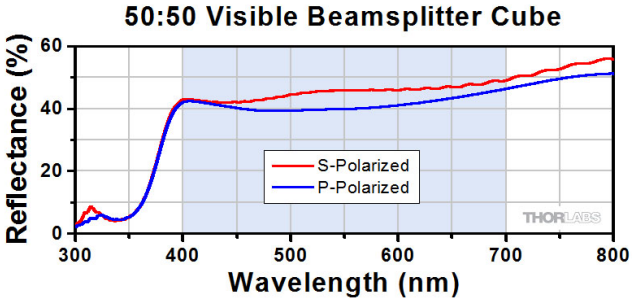


Figure 11: THORLABS data on 50:50 visible beamsplitter cube typical reflectance performance [22].

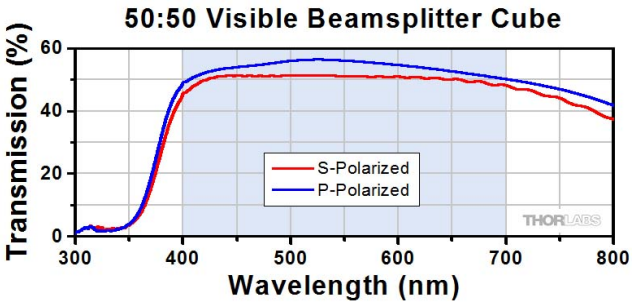


Figure 12: THORLABS data on 50:50 visible beamsplitter cube typical transmission performance [22].

It should also be noted that the beamsplitter is only nominally 50:50, as Figure 11 and Figure 12 show the light signal is not actually split into perfect halves. However, because this research does not involve the sort of photometry that is sensitive to polarization or specific light curves [23], this is not expected to negatively impact the data or results.

3.1.1.3 QHY174M-GPS

The frame-based sensor used for this research was a QHY174M-GPS CMOS sensor.

Metric	Value
Sensor Dimensions	11.25×7.03mm
Sensor Size	1920×1200pixels
Pixel Size	$5.86 \times 5.86 \mu m$
FOV (as configured)	$1^{\circ}12.97' \times 45.60'$
Spatial Resolution	$2.28''$/pix
Temporal Resolution	GPS $\pm 1 \mu s$

Table 3: QHY174M-GPS Specifications [24]

A keen observer would, understandably, be confused as to why an imaging sensor has a specification listed for temporal resolution. This is because the QHY174M-GPS includes an externally mounted Global Positioning System (GPS) receiver to provide very accurate location and timing data to the metadata of image files that it produces. This includes global shutter timing data down to the microsecond and data on the error between the system clock and the GPS time signal, such that the data points produced by both the CMOS and the EBC can be calibrated to a globally accepted time signal.

Given how the overall objective of this research is to investigate how the increased temporal resolution per data point of an EBC can make up for its lower spatial resolution than the CMOS sensor, the QHY174M-GPS' ability to provide and share an accurate time signal with both sensors will be very important in the discussion of data processing and analysis later on in this document.

While the telescope and mount have belonged to AFIT for the past decade or so, this sensor technology coming to the commercial market is one of the significant developments from the serious hobbyist Commercial Off-The-Shelf (COTS) market over the past decade or so that enabled this research.

3.1.1.4 DVxplorer

The EBC used for this research was iniVation's DVXplorer high resolution event-based sensor.

Metric	**Value**
Sensor Dimensions	5.76×4.32mm
Sensor Size	640×480pixels
Pixel Pitch	$9\times9\mu m$
FOV (as configured)	$37.36' \times 28.02'$
Spatial Resolution	$3.5''$/pix
Temporal Resolution	200 μs

Table 4: iniVation DVXplorer Specifications [25]

A temporal resolution of 200 μs is coarser than that of the GPS time signal, 200 μs of temporal resolution for each event is significantly finer temporal resolution than frame-based sensors traditionally provide for data captured over the entire frame.

The EBC was also at a disadvantage in this setup because it had a smaller Field of View (FOV) than the CMOS sensor due to the EBC's smaller sensor area. While the CMOS sensor area could have been artificially cropped to the size of the EBC's sensor, this was not done because the focus of this research was to compare COTS sensor options and not explicitly to compare the two different types of sensors for the same sensor size.

3.1.2 Software

This subsection details the software that was used to conduct this research. All software used was COTS software that did not require any special privileges to obtain, and in most cases, the software was open-source as well. This subsection is not intended to assert that only this software can be used to perform the workflow described in this chapter, rather, it describes the software that was chosen for this research on the basis of availability and the author's familiarity with the software.

3.1.2.1 TheSky

TheSky Imaging Edition[1] is a professionally built astronomy software package that was used to command the AstroPhysics 1100 GTO mount to point the Takahashi FSQ-106ED telescope at areas of interest in the night sky in addition to providing sidereal tracking commands to the mount to maintain stars as point sources of light to be used for plate solving images.

3.1.2.2 SharpCap

SharpCap[2] was used to command the QHY174M-GPS CMOS sensor, and, critically, to set the system time on the laptop used for data collection to the GPS time signal.

[1] `https://www.bisque.com/product/thesky-imaging-edition/`
[2] `https://www.sharpcap.co.uk/`

3.1.2.3 DV Software

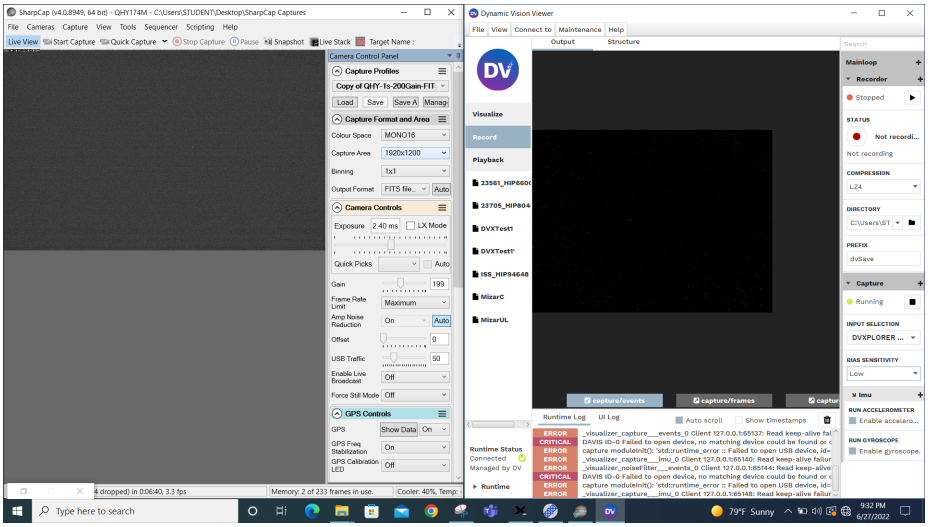


Figure 13: Screenshot showing typical research setup with SharpCap and DV software windows open and ready to record data.

iniVation's DV software[3] was used to command the DVXplorer EBC for data collection and to review collected data from recorded files prior to more intensive processing.

[3] `https://inivation.com/dvp/dvsoftware/`

3.1.2.4 Stellarium

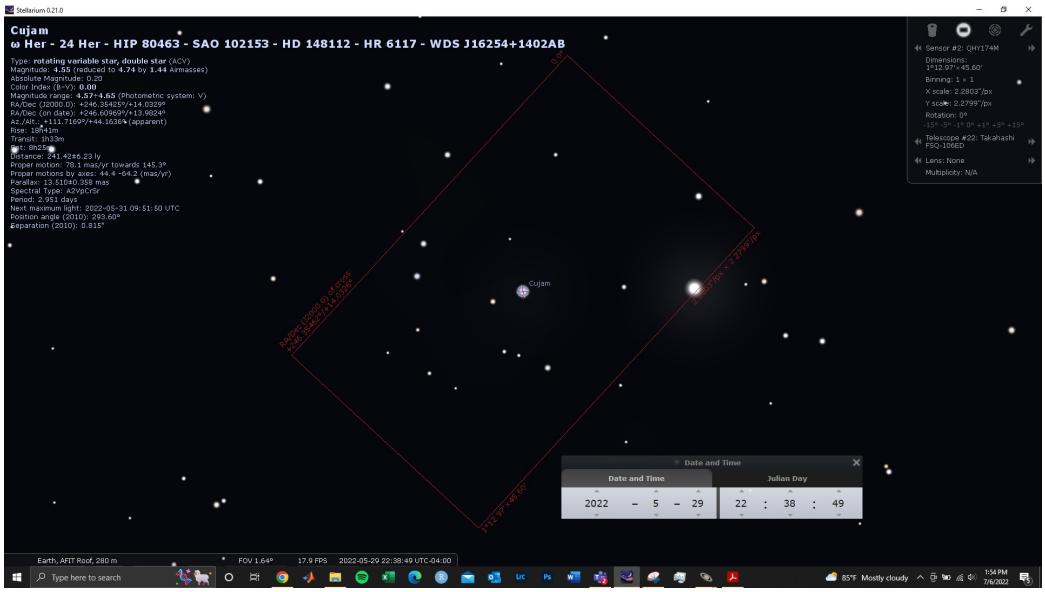


Figure 14: Screenshot showing Stellarium being used to predict the transit of NORAD 23705 by HIP 80463 through the FOV of the QHY174M-GPS. Compare to the actual transit shown in Figure 15.

Stellarium[4] is a free open source planetarium application for most common operating systems that was used to simulate the location of stars and Resident Space Object (RSO)s in the night sky relative to the Reactor Hill observatory. Prior to each night of data collection it was used to build an observation target package by predicting the passes of RSOs in front of stars that could be observed to gather position data on the RSOs of interest.

[4] `https://stellarium.org/`

3.1.2.5 AstroImageJ

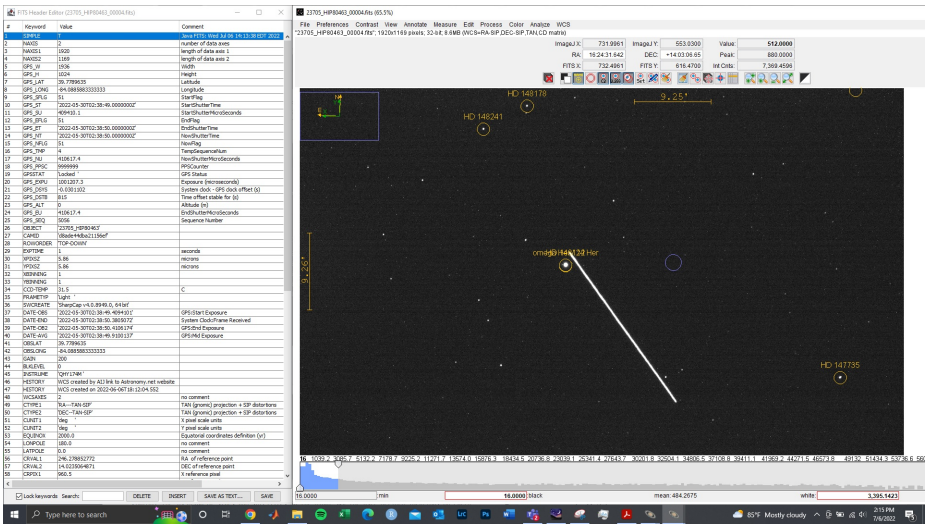


Figure 15: Screenshot showing AstroImageJ displaying a plate solved image of the transit of NORAD 23705 by HIP 80463, along with the metadata in the FITS header. Compare to the predicted transit shown in Figure 14.

AstroImageJ (AIJ)[5] was used to open the FITS files produced by the QHY174M-GPS and process the star fields into plate solutions via the Astrometry.net[6] bridge. By handling the plate solving the FITS files and providing easy access to the World Coordinate System (WCS) and SIP data in the FITS header, AIJ allowed for the translation from the pixel space of the CMOS sensor's images to the right ascension and declination space of the night sky. It was this translation that allowed stars, and by extension data points from streaks made by RSOs, to be translated from pixels to position angles.

Only introduced as a separate version of ImageJ in 2017 [26], AIJ is one of the significant recent developments from the serious hobbyist and academic world that enabled this research.

[5] https://www.astro.louisville.edu/software/astroimagej/
[6] https://nova.astrometry.net/

3.1.2.6 Spyder IDE

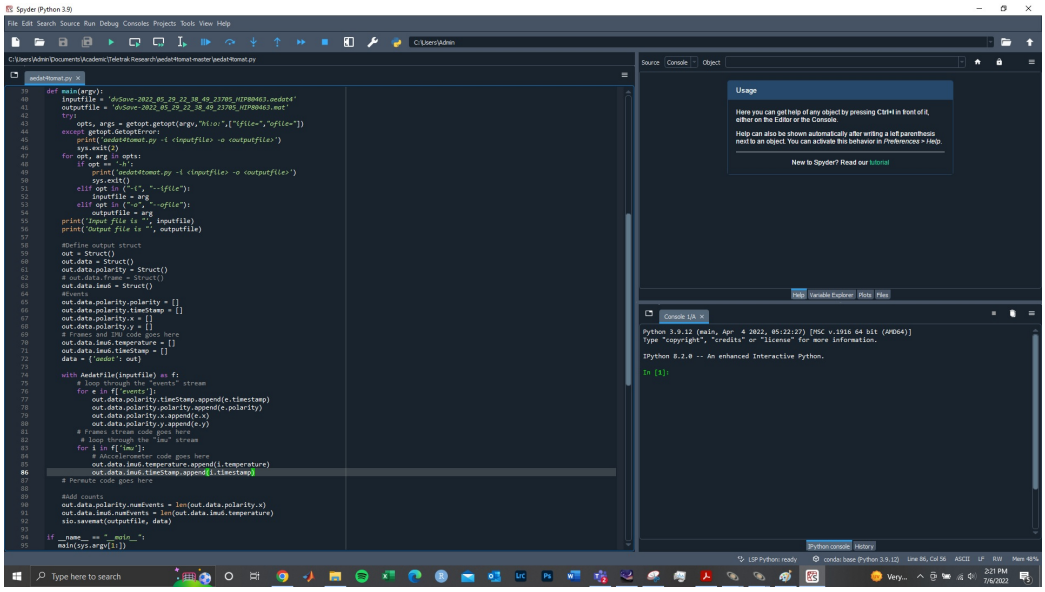


Figure 16: Screenshot showing aedat4tomat.py [27] in the editor pane of the Spyder IDE.

The Spyder[7] Python Integrated Development Environment (IDE) was used to run the aedat4tomat.py [27] file that was used to convert the data produced by the EBC into .mat files that could be accessed by Matlab.

3.1.2.7 Matlab

Matlab[8] [28] was used to process data points from the DVXplorer, in the form of .mat files, into data point clouds from which specific data points could be extracted from the EBC pixel space. Matlab was also used to apply the WCS and SIP data from FITS headers to translate data points from each sensor into right ascension and declination [29].

Matlab was then used to process the right ascension, declination, and time data from each sensor into an orbit update for the RSO of interest.

[7] https://www.spyder-ide.org/

[8] https://www.mathworks.com/products/matlab.html

3.1.2.8 Microsoft Excel

Excel[9] was used to model the translation between the EBC pixel space and the CMOS pixel space described in Section 3.3.2.2 in addition to being used to store both raw and processed data.

3.2 Collection Techniques

3.2.1 Location

Figure 17: Google Maps image of observation site location

All data was collected from AFIT's Reactor Hill Observatory, Wright-Patterson AFB building 470 J, at a location of 39.77896° north and 84.08859° west.

3.2.2 Preparation

While nights suitable for observing could be expected a few days in advance, the decision of whether or not to make observations ultimately became a day-of decision because of the need to rely on the most up to date forecast and Two-Line Element (TLE) data possible.

[9] `https://www.microsoft.com/en-us/microsoft-365/excel`

To determine if a night was suitable for observing, astronomy-oriented weather forecasts such as Clear Dark Sky[10] were consulted to determine if the forecasted cloud cover would prevent observations of stars and RSOs that night.

If a night was deemed suitable for observations, the latest TLE file for objects of interest that could be detected by the EBC were downloaded from Celestrak[11] and loaded into Stellarium to simulate the orbits with respect to AFIT's Reactor Hill observatory location.

Stellarium was then used to build an observation target package of RSOs that would pass within the field of regard of the observatory in the time between sunset and approximately 2300L. This was done by zooming Stellarium's view out to a fisheye view, speeding up time, and watching for RSOs crossing the screen. Each RSO observed crossing the screen was then selected and tracked across its path until it crossed near a star brighter than about 7th visual magnitude surrounded by enough[12] stars to yield a plate solution from CMOS images. Using Stellarium's simulation of the EBC's FOV superimposed over the star field, the time at which the RSO would enter the FOV, the HIP catalog number of the central star, and the NORAD catalog number of the RSO were recorded into an observation target package similar to the one shown below to inform the pointing of the telescope and the naming of the data files produced by both sensors.

[10] `https://www.cleardarksky.com/c/Dayton_OHkey.html?1`

[11] `https://celestrak.org/NORAD/elements/gp.php?GROUP=visual&FORMAT=tle`

[12] While there was no specific criteria for "bright enough" or "enough stars", Figure 14 provides an example of the sort of star field that was desired. Generally, 10 or more stars are required [7], but plate solutions can be made with fewer.

NORAD ID	Observation Star (HIP)	Observation Time (Local)
15772	73199	21.16.23
26070	66234	21.17.56
25732	69899	21.27.20
28415	66763	21.29.21
20511	58287	21.30.46
21423	45038	21.34.06
23343	91139	21.39.53
23343	107648	21.41.57
⋮	⋮	⋮

Table 5: Observation Target Package Example for 30 June 2022

3.2.3 Data Collection

Once at the observatory, after connecting all sensors to the laptop used to collect data, the information from the observation target package was followed to generate observations of each RSO of interest transiting in front of a star.

This was done by using TheSky software to point the telescope at the star of interest prior to the transit and then waiting until just before the transit time to begin collecting data. At about 10 seconds prior to the expected transit time, SharpCap was used to set the laptop's system clock to the GPS signal, at about 7.5 seconds prior to the transit the CMOS was commanded to begin recording frames via SharpCap, and at about 5 seconds prior to the transit, the EBC was commanded to start recording events via the DV software.

This method typically yielded 3 to 5 CMOS frames and 1 to 2 seconds of useful EBC data, with a few seconds of data on either side of the transit as well.

3.3 Data Processing

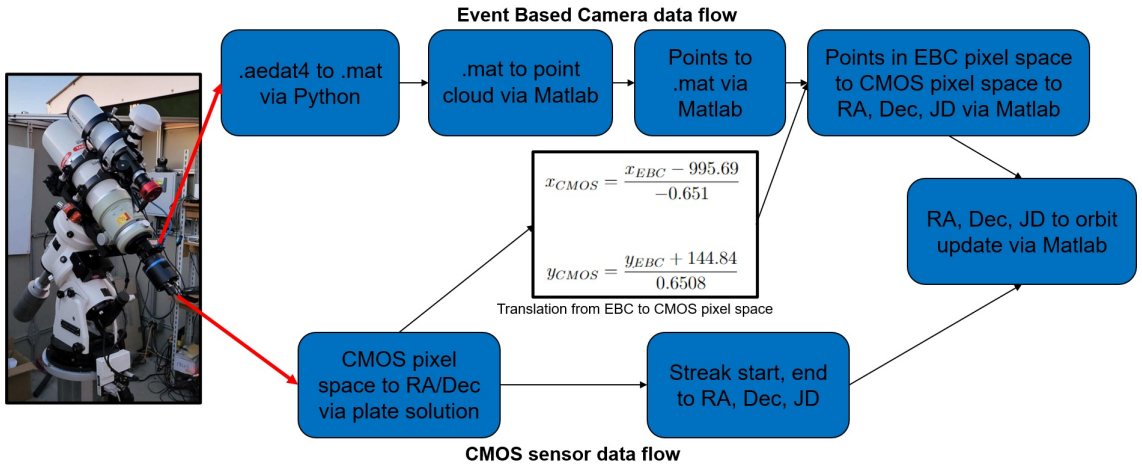


Figure 18: Depiction of the workflow used to process data for each sensor.

Data from each sensor was processed using the workflow shown in Figure 18, the steps of which will be detailed in this section.

3.3.1 CMOS Sensor Data Processing

For each set of images from an observed transit, all but one of the images without an RSO streak were deleted. The remaining images were then loaded into AIJ and plate solved via the bridge to Astrometry.net to populate each file's FITS header with WCS and SIP data for the frame.

AIJ was then used to find the location of the beginning and end point of each streak in the FITS pixel space, such that a .xlsx file could be populated with data on each point's x position, y position, and time of capture based on the image metadata.

This file was then imported into the Matlab CMOS data conversion tool along with a plate solved FITS image to translate each point from the CMOS pixel space into right ascension and declination values based on the WCS and SIP data from the plate solution. This right ascension, declination, and time data file was then input

into the orbit update model to generate the orbit update.

The remaining image file without a streak from each transit was preserved as a control image of the star field, to be used for troubleshooting in the event that Astrometry.net was unable to generate a plate solution for any of the images with an RSO streak. Additionally, it could be used as a single source of FITS header data to provide constant WCS and SIP values for each frame of CMOS data from a transit to simplify the processing required to translate from the CMOS pixel space into right ascension and declination.

Right Ascension (°)	Declination (°)	Julian Date
159.3255	31.4213	2459760.62145329
159.3231	31.4228	2459760.62145330
159.0366	31.6493	2459760.62146489
159.0346	31.6509	2459760.62146490
158.7480	31.8776	2459760.62147649
158.7462	31.8787	2459760.62147651
158.4587	32.1050	2459760.62148809
158.4564	32.1067	2459760.62148811

Table 6: Example of CMOS Data from NORAD 23343 HIP 51840 Transit, 6-29-2022

3.3.2 EBC Data Processing

The processing of data from the EBC is more involved than that of CMOS sensor data because the EBC data can only be processed into an orbit update after being pulled out of the data point cloud and translated into the CMOS pixel space. This limitation exists for EBC data because, without the ability to plate solve EBC data, EBC data could not be translated directly into right ascension and declination. First,

the EBC data was processed from its native .aedat4 file format to the .mat file format for use in Matlab by way of the aedat4tomat.py [27] code, which selectively wrote time and pixel data to a .mat file while omitting accelerometer and temperature data to minimize the size of the .mat file. Next, the EBC data was further reduced by ignoring events with negative polarity so that only positive events generated by pixels being exposed to increasing brightness were considered.

Figure 19: An example of the EBC's pixels taking time to settle after an RSO passes, visible as the red trail behind the RSO.

This was done because, as Figure 19 shows, pixels that a bright object had recently passed through still produce negative event data while taking time to settle. Including negative events in the analysis would yield erroneous data for the RSO's position, so they were omitted from EBC data analysis.

3.3.2.1 Noise Reduction

The remaining data from the .mat file was then processed into a point cloud in the form of $[x, y, t]$ data so that the streak of an RSO pass could be visualized in the

point cloud, albeit with significant noise, as shown in Figure 20.

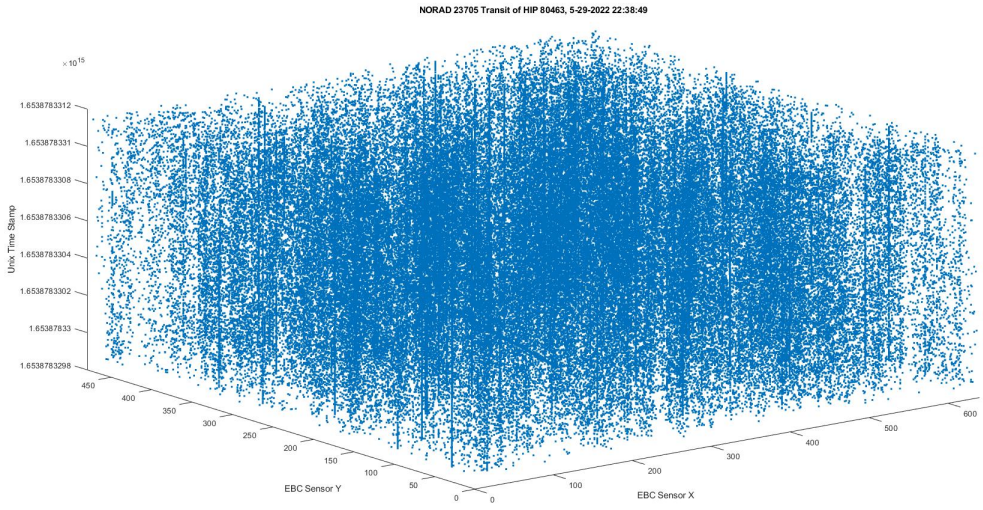


Figure 20: An example of the raw point cloud generated from all EBC positive events. The streak generated by the RSO is barely visible near the lower center of the point cloud.

The exact source of the noise in unprocessed EBC data is is not precisely known, but the noise is likely attributable to several factors, such as pattern noise, stray light, or defective pixels.

Because of how difficult this noise made it to select data points from the streak in the raw data point cloud, a histogram of the counts for each pixel like the one shown in Figure 21 was used to highlight the location of streaks within the EBC's pixel space.

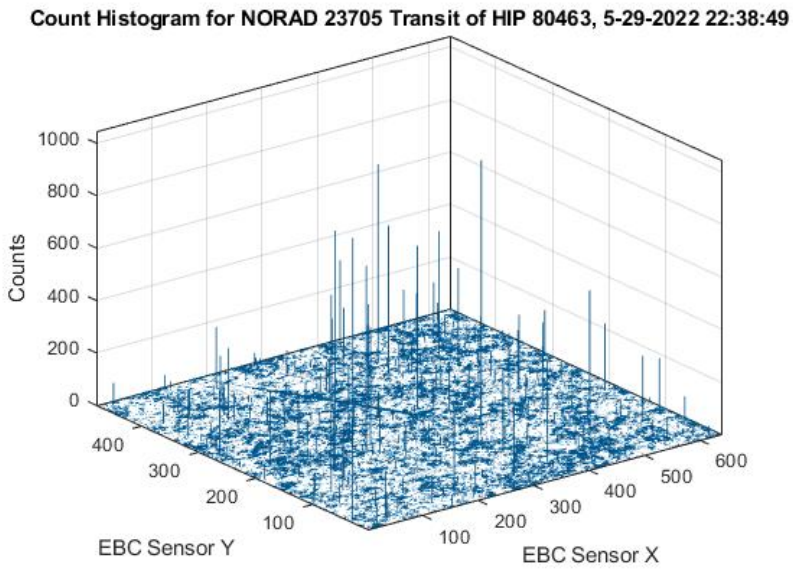


Figure 21: An example of a histogram of counts from the point cloud generated from EBC data. The streak generated by the RSO is visible on the left side of the histogram.

Histograms like this were used to provide information on where in the EBC's pixel space the streak occurred, and on how much cropping of the point cloud could be done to best isolate the streak.

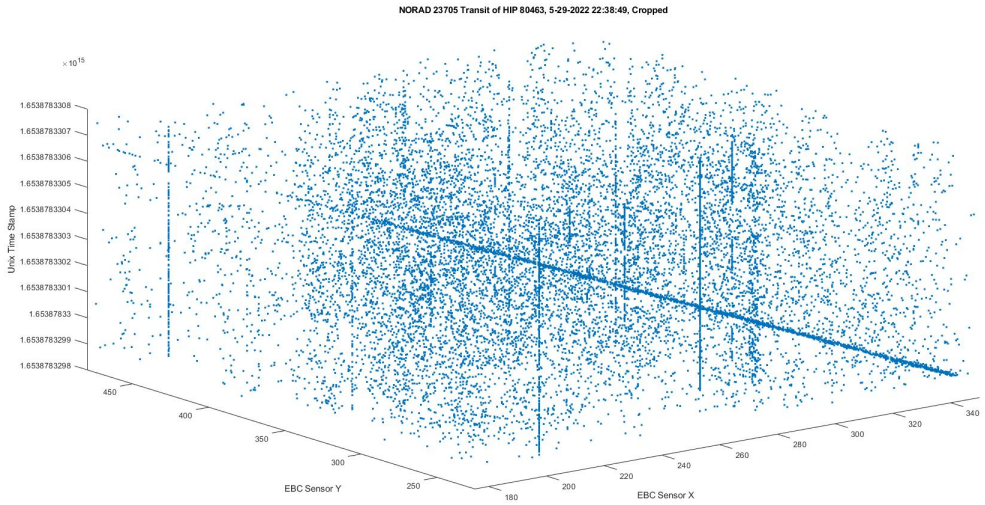


Figure 22: An example of a cropped point cloud of EBC data. Notice that the bounds of each axis are smaller than those shown in Figure 20.

A cropped noise cloud like the one shown in Figure 22 much more clearly isolates the streak than the point cloud in Figure 20 by cutting away the parts of the point cloud that do not contain the streak.

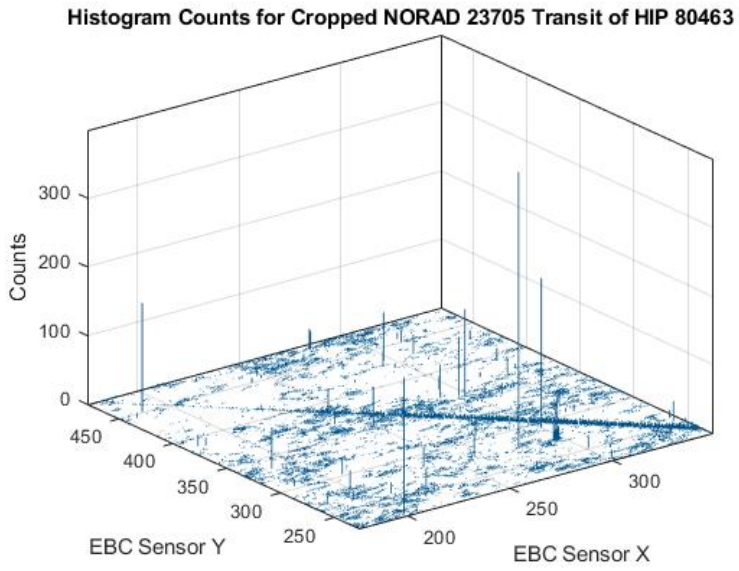


Figure 23: An example of a histogram from the cropped point cloud.

The histogram from this cropped data, shown in Figure 23, was further used to provide information on the noise counts in the cropped data such that a band pass filter could be applied to remove noise from the point cloud without significant loss of data. To implement this, all pixel locations in the histogram that generated fewer than a lower limit or more than an upper limit of events were removed from the point cloud to cut out random noise and hot pixels.

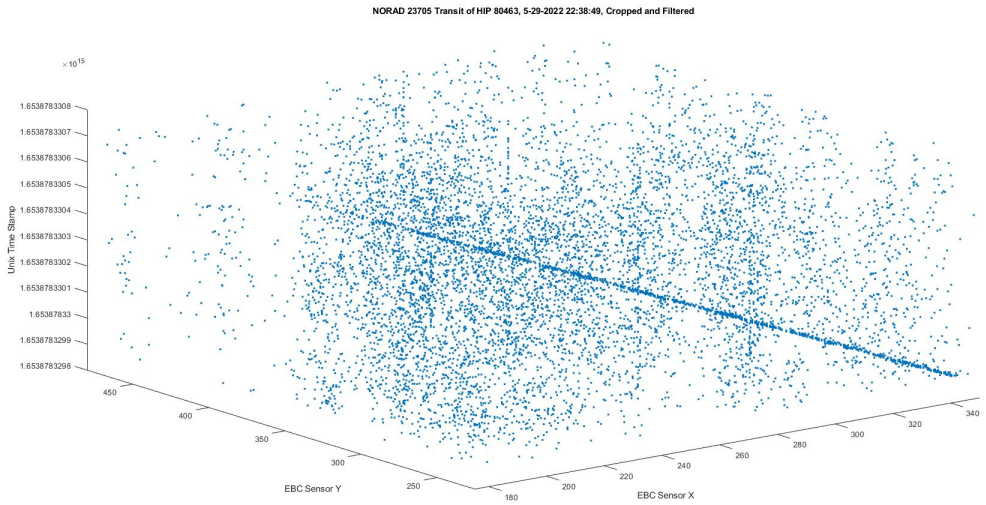


Figure 24: An example of a cropped and filtered point cloud.

This cropped and filtered data was then passed to a program that implemented the Hough transform [30] to automatically find the streak in the data, as shown in Figure 25:

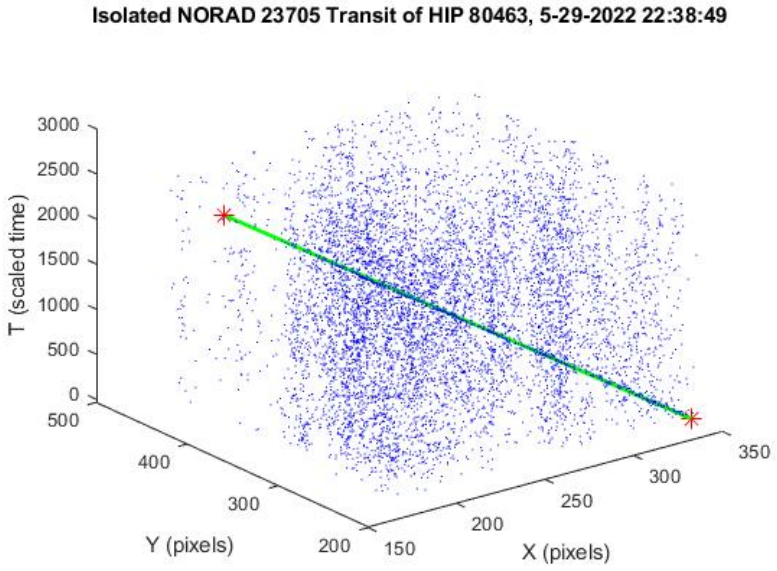


Figure 25: An example of a streak detected in a data point cloud by the Hough transform.

The data shown in Figure 25 could then be automatically selected using a random sample consensus algorithm to form an analytic representation of the streak. While the ability to automatically extract streaks from the data point cloud was demonstrated for this research, the best fit model generated by the algorithm was observed to occasionally cover only part of the streak or to slightly mis-represent the streak by a small angle. Because of this, points were selected manually from the streak, which were saved into a .mat file to be processed with FITS header data into a .xlsx file as right ascension, declination, and Julian date data. This data was then finally processed by Matlab to generate an update of the RSO's orbit.

3.3.2.2 Pixel Space Translation

Because the EBC was not sensitive enough to produce data containing enough stars to yield a plate solution that would permit a direct translation to right ascension and declination, the pixel space of the EBC was mapped to the pixel space of the CMOS as an intermediate step before ultimately being translated to the right ascension and declination space using the same WCS and SIP algorithms that were used for CMOS data.

To map the pixel space of the sensors to each other, 9 data points of a known star were recorded on both cameras from the corners of the EBC's pixel space, the middle of each side of the pixel space, and the center of the pixel space. The data from each point from each sensor was then input into a spreadsheet to calculate a regression between the two sets of data.

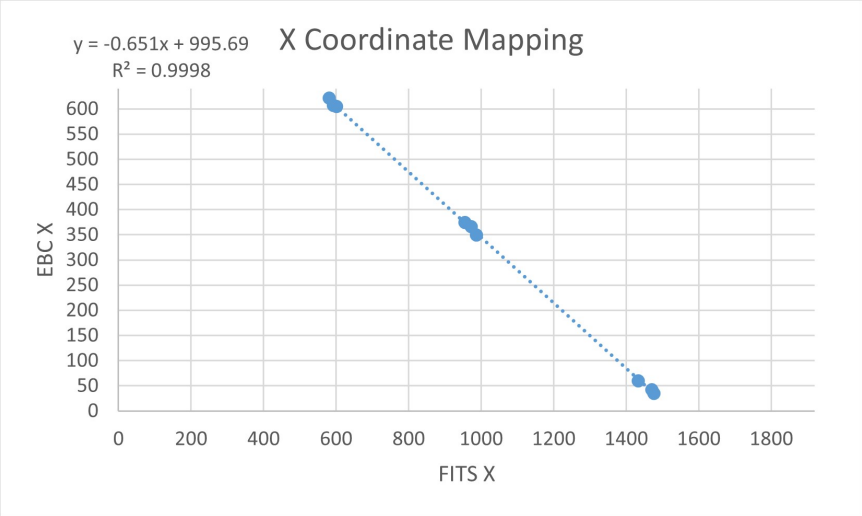


Figure 26: The plot used to generate the mapping from the pixel space of the EBC to the pixel space of the CMOS in the X axis.

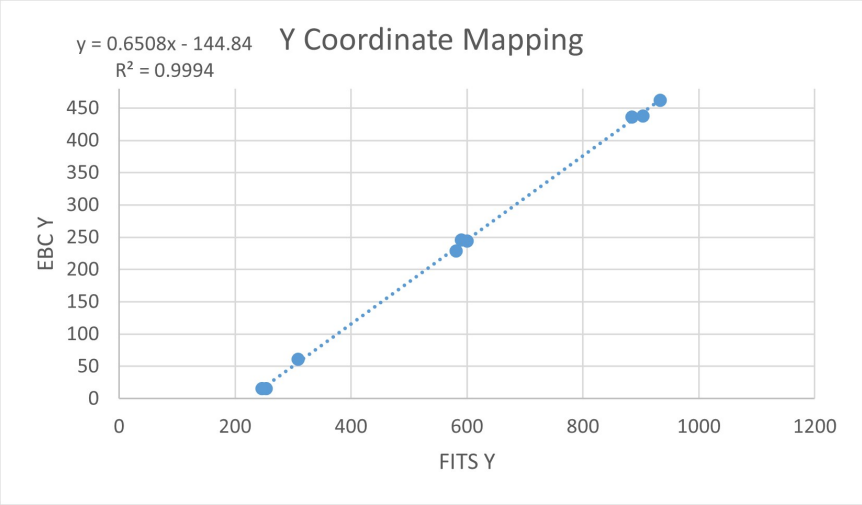


Figure 27: The plot used to generate the mapping from the pixel space of the EBC to the pixel space of the CMOS in the Y axis.

The best fit generated by this approach was a linear regression that was used to map the EBC pixel space to the CMOS pixel space, as shown in Equation (3) and Equation (4):

$$x_{CMOS} = \frac{x_{EBC} - 995.69}{-0.651} \tag{3}$$

$$y_{CMOS} = \frac{y_{EBC} + 144.84}{0.6508} \tag{4}$$

Nonlinear fits were considered, but linear regression was selected as the best possible translation between the two pixel spaces because a nonlinear translation between the two parallel and rectilinear spaces would not be logical. The linear model also offers simple equations that each achieve a high R^2 value, where the primary source of error is the result of the difference in pixel sizes between the two sensors preventing a clear match between the center of points in each pixel space.

3.3.3 GPS Time Correction

An aspect of data processing that is conspicuously missing, given the use of a time signal from the QHY174M-GPS, is correction for the offset between the system time assigned to each data point and the true GPS time provided by the GPS receiver based on the offset data that accompanies each frame. This correction was not performed because correcting data based on the time signal in the metadata of each CMOS frame would have improved the CMOS data more than it would have improved the EBC data in a manner that would interfere with the intent of a true comparison between these two sensors.

While using SharpCap to sync the system time to GPS time at one point shortly before collecting data provides a baseline of a very accurate system time, the synchronization is not maintained continuously and limitations of the laptop's internal clock allow for some timing error to accumulate seconds later as data is being collected. This is a known limitation, so SharpCap provides the offset between system time and GPS time in the FITS header of each frame so that any included data points can be calibrated to the GPS time signal.

The problem is that these offsets vary in both magnitude and polarity between

frames from the same data capture, such that a constant offset cannot be applied to all data from that capture. The EBC does not have any comparable timing offset data, so calibrating the frame-based CMOS data to GPS time while performing an approximate calibration to the EBC data points based on the frames during which they were collected would introduce a source of differential error between the two measurements.

It is impossible to know how much this would affect the difference in results between the two sensors, so rather than correcting one sensor better than the other, all data was left uncorrected from its respective metadata to allow the system clock error to affect both sensors equally.

3.3.4 Orbit Update Calculation

The orbit update for each dataset was produced using a non-linear weighted least squares approximation [31] while using finite differencing [13] to estimate the partial derivatives of the system.

First, once the observation data was read into the program, the residual was calculated between each observed right ascension (α) and declination (δ) and the values of α and δ that were expected based on propagating the initial TLE to the time of each data point. The standard deviations for the residuals of α and δ were then used to construct the weighting matrix W for n data points from a single observation:

$$W = \begin{bmatrix} \frac{1}{\sigma_{\alpha 1}^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_{\delta 1}^2} & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_{\alpha n}^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sigma_{\delta n}^2} \end{bmatrix} \tag{5}$$

In the absence of well characterized sensor noise and bias values, this was done to approximate the noise in each set of data points from an observation such that the right ascension and declination can be properly weighted relative to each other for the orbit update.

The state calculated from the TLE was then propagated to the time of each observation to again calculate the residuals for each α and δ measurement. This time, the data was assembled into the b matrix of residuals:

$$b = \begin{bmatrix} \alpha_{o,1} - \alpha_{c,1} \\ \delta_{o,1} - \delta_{c,1} \\ \alpha_{o,2} - \alpha_{c,2} \\ \delta_{o,2} - \delta_{c,2} \\ \vdots \end{bmatrix} \tag{6}$$

This was done to quantify the residual between the observed α and δ values and the expected values calculated by propagating the state to each data point. As the differential correction iterates through different updates to the state, this b matrix is used as a measure of the accuracy of the updated state to drive the next update of the state described in Equation (10) or to indicate that the updated state has converged based on Equation (13).

Next, this process of propagating to each data point was repeated, but with the added step of iterating through each of the six elements of the state vector and adding an additional 1×10^{-12} of its current value to the element before propagation:

$$\hat{X} = \hat{X} + \hat{X} \times 10^{-12} = \hat{X} + \Delta_i \tag{7}$$

where the change that resulted from this small perturbation was divided by the value of the perturbation to simulate a partial derivative:

$$\frac{\partial \text{observations}}{\partial \hat{X}} \approx \frac{f(\hat{X} + \Delta_i) - f(\hat{X})}{\Delta_i} \tag{8}$$

The A matrix of partial derivatives was then built over the course of successive adjustments to each element of the state vector for each observation via the method of finite differencing:

$$A = \frac{\partial \text{observations}}{\partial \hat{X}} \tag{9}$$

where the partial derivatives at each point were used to build the columns of A, with a column for each of the six elements of the state vector.

Because a partial derivative fundamentally describes the change in one value with respect to the change in another value, the alternative used here to avoid going through the algebra required to find the true partial derivative matrix is to "take small differences of the state to determine their effect on the system" [13]. Or, in other words, to perform a mathematically intense guess and check in such a manner that the partial derivatives can be found numerically rather than analytically. This is how the method of finite differencing earns its name.

With the A and b matrices constructed, they are then used with the sensor weighting matrix to produce the update to the state:

$$\delta\hat{x} = (A^T W A)^{-1} A^T W b \tag{10}$$

which is then added to the state:

$$\hat{X}_n = \hat{X}_{n-1} + \delta\hat{x} \tag{11}$$

This process is repeated to create another b matrix of residuals and another A matrix of partial derivatives until the root mean square of the residuals converges to

set value, where the RMS is calculated from the residuals matrix b, the weighting matrix W, and the number of observations N:

$$RMS = \sqrt{\frac{b^T W b}{N}} \tag{12}$$

$$\left| \frac{RMS_{old} - RMS_{new}}{RMS_{old}} \right| \leqslant 0.001 \tag{13}$$

At which point the state estimate can be used for analysis. Values for the convergence criterion too large could cause the solution to converge prematurely, while values too small could cause the program to iterate for an unnecessarily long time between estimates that are good enough or make a jump to an unrealistic solution and fail to converge entirely. In this case, the criterion for convergence was defined as 0.001 based on observations that 0.01 was too high but 0.0001 could cause convergence issues for datasets that successfully converged for a value of 0.001.

3.3.5 State Propagation Calculation

State propagations between two epochs were handled with analytical solutions [32] for this research, based on values for the six Classical Orbital Elements (COE)s and associated perturbations contained in the TLE.

While the orbit update calculations used the six elements of a state based on three components each from an RSO's position and velocity vectors, the calculation between points in time were performed using COEs due to their relative simplicity when compared to more complicated methods of orbit determination that rely on position and velocity vectors (such as Cowell's method) and due to the limited availability of professional orbit determination software. It is understood that working in terms of COEs is a low fidelity approach, however, this is considered acceptable for the

relative comparisons made for this research. Additionally, because propagator error never increased above 1 $\frac{km}{day}$, any propagator errors introduced by working with COEs were deemed negligible on the scale of errors inherent to orbit updates made by a single observer at a single location.

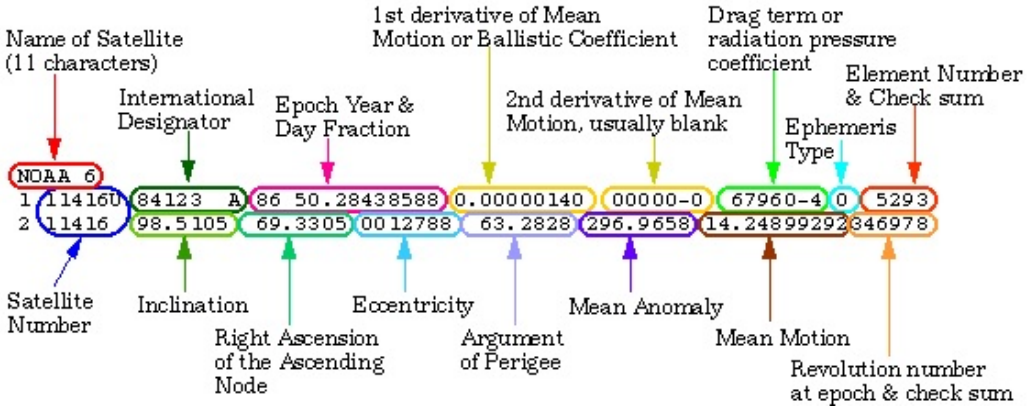


Figure 28: Example of a TLE, indicating the locations of state data within the TLE.

Beginning with data from a TLE in the format shown in Figure 28, the first derivative of mean motion, inclination, right ascension, eccentricity, argument of perigee, mean anomaly, and mean motion were extracted. The inclination, eccentricity, mean motion, and first derivative of mean motion were then used to calculate the rates of change of right ascension, argument of perigee, and eccentricity to account for the circularizing effect of drag and the precession effect of J_2:

$$\dot{e}_{drag} = \frac{-2(1-e_0)\dot{n}_0}{3n_0} \tag{14}$$

$$p_0 = (1-e_0^2)\sqrt[3]{\frac{\mu}{n_0^2}} \tag{15}$$

$$\bar{n} = n_0[1 + \frac{3}{2}J_2(\frac{R_\oplus}{p_0})^2\sqrt{1-e_0^2}(1-\frac{3}{2}\sin^2(i_0))] \tag{16}$$

$$\dot{\Omega}_{J_2} = [-\frac{3}{2} J_2 (\frac{R_\oplus}{p_0})^2 \cos(i_0)]\bar{n} \tag{17}$$

$$\dot{\omega}_{J_2} = [\frac{3}{2} J_2 (\frac{R_\oplus}{p_0})^2 (2 - \frac{5}{2}\sin^2(i_0))]\bar{n} \tag{18}$$

Using these rates of change and the known change in time between two epochs of interest, Δt, the state of the RSO was then propagated to the future epoch using following equations:

$$n = n_0 + \dot{n}_0 \Delta t \tag{19}$$

$$e = e_0 + \dot{e}_{drag} \Delta t \tag{20}$$

$$\Omega = \Omega_0 + \dot{\Omega}_{J_2} \Delta t \tag{21}$$

$$\omega = \omega_0 + \dot{\omega}_{J_2} \Delta t \tag{22}$$

$$M_f = M_0 + n_o \Delta t + \frac{\dot{n}}{2} \Delta t^2 \tag{23}$$

where the value of the mean anomaly, M, was iterated using the Newton-Raphson method to find the eccentric anomaly E such that ν could be calculated:

$$E_n = E_{n-1} + \frac{M_f - E_{n-1} + e\sin(E_{n-1})}{1 - e\cos(E_{n-1})} \text{ while } |E_n - E_{n-1}| > 10^{-9} \tag{24}$$

$$\nu = 2\tan^{-1}(\sqrt{\frac{1+e}{1-e}} \tan\frac{E_f}{2}) \tag{25}$$

These values were then used to calculate the position vector from the center of the Earth to the RSO at the epoch of interest in the perifocal frame:

$$p = (1 - e^2)\sqrt[3]{\frac{\mu}{n^2}} \tag{26}$$

$$R_{PQW} = \frac{p}{1 + e\cos\nu}[\cos(\nu)\hat{P} + \sin(\nu)\hat{Q}] \tag{27}$$

Rotation matrices were then used to translate the position vector from the perifocal frame to the Earth-Centered Inertial (ECI) frame:

$$R_{IJK} = [Rot_3(-\Omega)][Rot_1(-i)][Rot_3(-\omega)]R_{PQW} \tag{28}$$

This R vector for the RSO was then used to calculate the right ascension and declination of the RSO.

3.4 Right Ascension and Declination Calculation

To find the right ascension and declination angles between the center of the Earth and the RSO, the ECI position vector of the observing site had to be established as a function of longitude λ, latitude L, altitude, and Greenwich Sidereal Time (GST). While the location of the Reactor Hill observatory is well characterized in an Earth-fixed frame of reference, the ECI position of the observing site changes due to the rotation of the Earth:

$$x = \left| \frac{a_\oplus}{\sqrt{1 - e^2\sin^2(L)}} + \text{alt} \right| \cos(L) \tag{29}$$

$$z = \left| \frac{a_\oplus(1 - e^2)}{\sqrt{1 - e^2\sin^2(L)}} + \text{alt} \right| \sin(L) \tag{30}$$

$$R_{site} = \begin{bmatrix} x\cos(GST + \lambda) \\ x\sin(GST + \lambda) \\ z \end{bmatrix} \tag{31}$$

where this site position vector can be used to find the vector from the observation site to the RSO, ρ. This ρ vector was then translated into the South-East-Zenith (SEZ) topocentric frame:

$$\rho_{SEZ} = [Rot_2(\frac{\pi}{2} - L)][Rot_3(GST + \lambda)]\rho_{IJK} \tag{32}$$

This ρ_{SEZ} vector was then used to calculate the target's azimuth and elevation:

$$az = \sin^{-1}(\frac{\rho_Z}{\bar{\rho}}) \tag{33}$$

$$el = \tan^{-1}(\frac{\rho_E}{-\rho_S}) \tag{34}$$

which were then used to calculate the declination and right ascension of the RSO:

$$\delta = \sin^{-1}(\sin(el)sin(L) + cos(el)cos(L)cos(az)) \tag{35}$$

$$\alpha = \tan^{-1}\left(\frac{-(\sin(az)\cos(el)\cos(L))(\cos(\delta)\cos(L))^{-1}}{(\sin(el) - \sin(L)\sin(\delta)(\cos(\delta)\cos(L))^{-1}}\right) \tag{36}$$

These values for geocentric declination and right ascension were used to describe the position of the RSO relative to the center of the Earth, based on the position vector of the RSO at the epoch of interest.

It is true that the translation from the position vector to right ascension and declination could have been made more directly, but this method was chosen due to its ability to calculate the range, azimuth, and elevation from the observing location

to the target that can be useful for both debugging code and future research.

3.5 Results Calculation

For each updated state based on observation data, the accuracy of the updated state was compared to the TLE that was used to generate the observation and the future TLE from the day after the observations. Because both TLEs were generated using data from the Space Surveillance Network (SSN), both were considered to be truth data for the purpose of analysis.

The first result to be generated was the linear offset in kilometers between the RSO's true position vector at the epoch of the future TLE and the position vector calculated by propagating the updated state from the epoch of the first TLE to the epoch of the future TLE. The magnitude of the difference between the two position vectors, itself another vector, was then calculated and used to describe the separation between the RSO's true position and where it was expected to be based on the epoch.

A similar approach was taken to calculate the difference in kilometers per second of the velocity vectors between the calculated velocity at the epoch of the future TLE and the velocity calculated by data from the TLE itself.

These one dimensional residuals were then divided by the duration between the timestamp of the last data point and the epoch of the future TLE to calculate drift rates, or error accumulation rates, for each updated state. This was done because, while one dimensional position and velocity offsets quantify the accuracy of the updated state at the future epoch, they do not account for the time in which the error could increment such that the linear offsets of two cases with different propagation times are not directly comparable. Calculating drift rates solves this issue to provide a baseline value with which to compare accuracy between cases.

The accuracy of the EBC and CMOS state updates from each case were also

compared to each other on a relative basis, where the EBC error was divided by the error of the state from the CMOS data to produce a figure of merit to describe the relative performance of each sensor in each case. This figure of merit was then correlated with the number of data points produced by each sensor, such that the relative accuracy in each case could be directly compared to the number of data points from each sensor in each case.

3.6 Sources of Error

While not every source of error can be identified and quantified due to this research taking place in the real world and using real hardware, there is still sufficient data to estimate the magnitudes of spatial and temporal error in both CMOS and EBC data.

3.6.1 Spatial Error

The largest single source of spatial error in this research is likely to be the uncertainty with which the streaks of a RSO is extracted from the sensor data. In the case of the CMOS sensor, which generates streaks that are often more than 5 pixels (or 14″) wide, the error is expected to be minimal because the brightness of pixels across the streak can be used to estimate the true center of the streak from processes like centroiding. This process uses the light curve of a bright point in an image to estimate the center of the point to sub-pixel accuracy [33]. As a result, while the streak introduces some uncertainty in the CMOS data, the true position of the RSO in the streak can be estimated to within about a pixel or about 2.28″ of the RSO's true position. Based on this, the upper limit of uncertainty introduced by the spatial resolution of the CMOS is expected to be about 2.28″ for each data point.

However, because the data provided by the EBC is binary in the sense that events either do or do not exist, EBC data cannot be used in processes like centroiding that

allow CMOS sensor data to achieve sub-pixel accuracy because there is no light curve available to estimate the RSO's position within a streak. Estimates of the RSO's path through the point cloud like the Hough transform shown in Section 3.3.2.1 can be used to inform the selection of points within a streak, but without better spatial resolution from smaller EBC pixels the spatial uncertainty for each point cannot be less than 3.5″. A perfection of the algorithm demonstrated in Section 3.3.2.1 could solve this problem, but because the algorithm was not observed to consistently produce perfect results, it was not implemented for this research.

The mapping of the EBC pixel space to the CMOS pixel space also introduces error to the EBC right ascension and declination spatial data in the form of error between the modeled translation and the actual translation between the two pixel spaces. At its worst, this error is up to 23.5″ in the corners of the EBC frame, with a more tolerable 5.8″ in the center of the frame. However, due to how RSOs typically transit through the FOV of the EBC, the majority of data points are sourced from the center area of the frame where the error is in the range of about 5.8″ to 10.5″. While adding more data points to the dataset used to form the translation could have improved the translation's accuracy, the value of additional data points would still be limited by the fact that the EBC's spatial resolution is 3.5″ per pixel and the center points of bright stars used to make this translation would be uncertain.

Because these sources of error in the EBC data cannot be easily eliminated without further improvements in the spatial resolution of EBC technology, spatial data from the EBC should be considered less accurate even after translation than the spatial data from the CMOS sensor.

Another less significant source of spatial error is the light beam deviation caused by atmospheric turbulence, which introduces uncertainty of about 0.5″ to 2″ [16] to the light before it enters the telescope. The variation of errors from this seeing effect

is actually the basis for the EBC's ability to generate events from bright stars while tracking at the sidereal rate, so in this case it is a useful source of error that is intentionally not mitigated.

3.6.2 Temporal Error

The primary source of temporal error in this research is the drift of the system clock from the GPS time signal in the few seconds between using SharpCap to synchronize the time signals and the beginning of data collection. The problem is that, while the temporal resolution of the QHY174M-GPS is within 1 μs of GPS time for the beginning and end of each frame [24] and the EBC can sample with a resolution of 200 μs [25, 34] for each event, the limiting factor that applies to the EBC data is still the system clock of the laptop used to collect data. This drift only amounts to a maximum of about 3 ms of system clock lag from the GPS signal, but because the explicit purpose of this research is to characterize the EBC's ability to compensate for its reduced spatial resolution with increased temporal resolution, any deviation from GPS signal is detrimental to the accuracy of each data point's time stamp.

However, for a theoretical RSO with a semimajor axis of 7,000 km and a mean motion of 0.0011 $\frac{rad}{s}$, the angle travelled over the course of 3 ms is only 0.67″. While not ideal, this error is well within the spatial resolution of both the EBC and the CMOS sensor, such that the spatial resolution of the EBC remains its limiting factor.

IV. Results and Analysis

Introduction

Using the processes described in Chapter III, the collected data points were processed to produce orbit updates for multiple passes of multiple different satellites. This chapter details the results of the research and provides brief clarifying details on the results.

It is worth repeating at this point that the objective of this research is *not* to provide data from a single observer at a single location that can compete with the Space Surveillance Network (SSN)'s data, but rather to relate orbit updates produced by the new Event-Based Camera (EBC) technology to those produced by traditional Complementary Metal-Oxide Semiconductor (CMOS) technology while considering a Two-Line Element (TLE) from the SSN as a source of truth data.

4.1 Numerical Results

4.1.1 Multiple Pass Updates

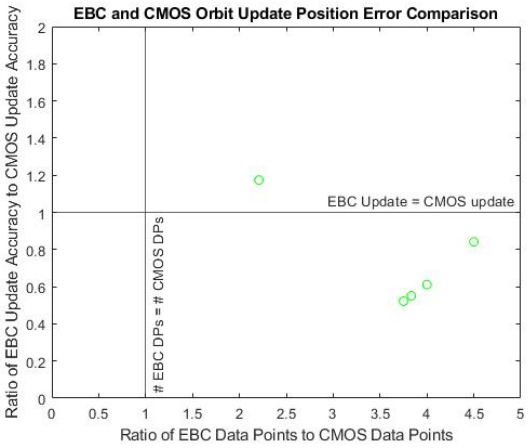
For orbit updates that were constructed over multiple observations, Section 4.1.1 details the magnitude of position error for each update:

Case	NORAD ID	Passes	EBC Error (km)	Points	CMOS Error (km)	Points
1	23705	2	30.116	45	15.723	12
2	23705	3	36.801	72	22.514	18
3	23705	2	29.377	46	16.194	12
4	23705	2	21.526	54	18.113	12
5	23343	2	17.94	53	21.056	24

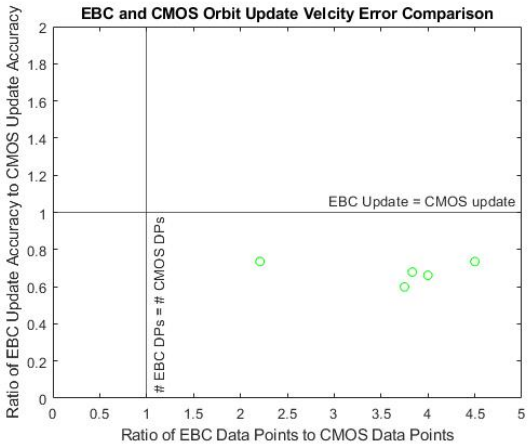
Table 7: Orbit Update Comparison from EBC and CMOS Data for Multiple Pass Cases

Where each orbit update was propagated to the epoch of a TLE published the day after the last set observations so that the magnitude of the R vector difference between the truth data and the orbit update could be calculated.

In each case, successive passes were collected during consecutive nights, for a total of three passes of data for NORAD 23705 and a total of two passes for NORAD 23343. To generate the multiple pass results for NORAD 23705 from 3 passes, Case 1 used data from the nights 1 and 2, Case 2 used data from nights 1, 2, and 3, Case 3 used data from nights 1 and 3, and Case 4 used data from nights 2 and 3.



(a) Position Error Comparison.

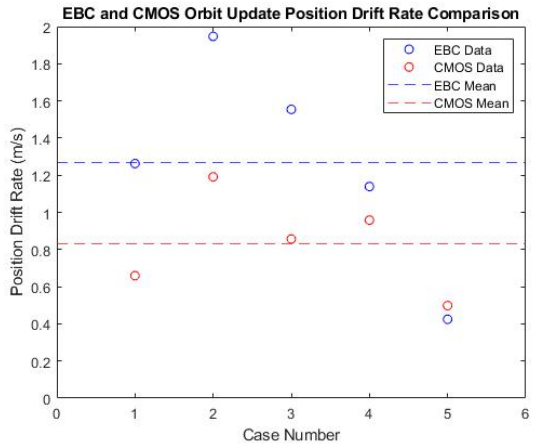


(b) Velocity Error Comparison.

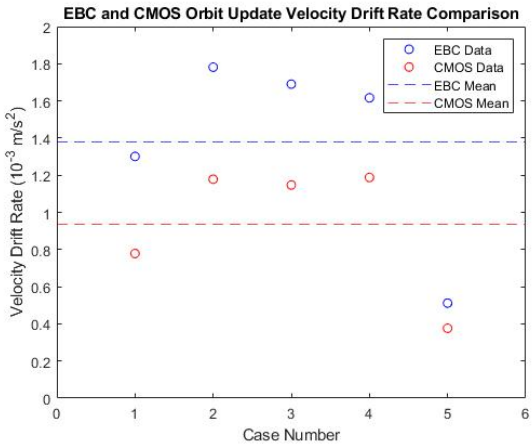
Figure 29: Plot comparing the relative number of EBC data points to CMOS data points and the corresponding difference in accuracy of the orbit update provided by each dataset.

Figure 29 indicates that, while the orbit updates produced from EBC data collected over multiple nights is accurate on the same order of magnitude as orbit updates from CMOS data, the orbit updates from the EBC are in most cases less accurate than the CMOS orbit update made from observing the same pass.

However, directly comparing the error in the R and V vectors at epoch does not account for the time over which the errors accumulate. Figure 30 shows a comparison of the error accumulation rates, or drift rates, for each case.



(a) Position Drift Comparison.



(b) Velocity Drift Comparison.

Figure 30: Plot comparing the drift rate magnitudes of the orbit update provided by the EBC and CMOS sensor for each case.

This data was calculated by dividing the magnitudes of the position and velocity vector errors by the duration between the timestamp of the last data point and the timestamp of the TLE used to make the comparison. While these error accumulation rates are in $\frac{m}{s}$ and $\frac{m}{s^2}$, this should not be taken to mean that the predicted position of the Resident Space Object (RSO) in question drifts at a constant rate from its true position. Rather, the magnitude of the errors were normalized by the time so that the error present in each case could be compared on an equal basis.

Figure 30 shows that, while the error from the EBC orbit updates are of a similar order of magnitude to those produced by EBC data, error from the EBC's update accumulates at a greater rate than that from the CMOS update. Once again, the EBC proves itself comparable to the CMOS sensor, but not better than the CMOS sensor.

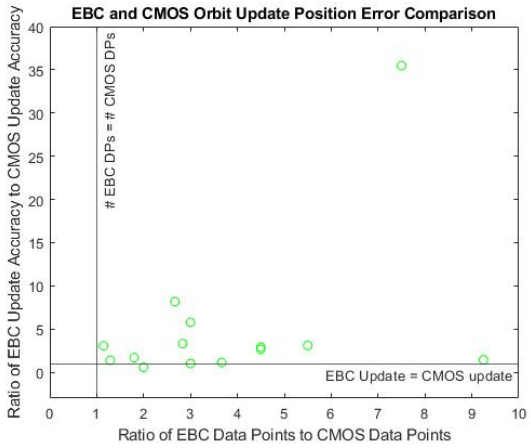
4.1.2 Single Pass Updates

For orbit updates that were constructed from a single observation, the following table details the magnitude of position error for each update:

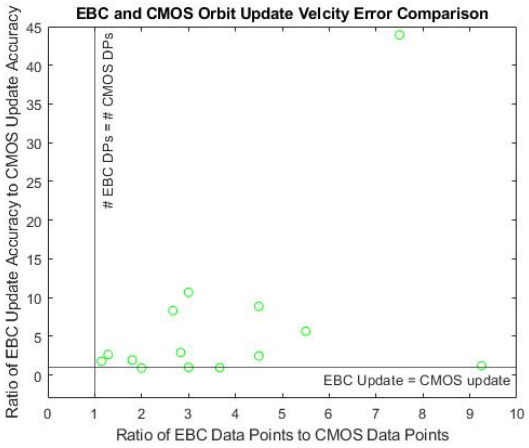
Case	NORAD ID	EBC Error (km)	Points	CMOS Error (km)	Points
1	23705	19.478	18	112.62	6
2	23705	14.153	27	41.079	6
3	23705	6.3727	27	16.861	6
4	23561	17.197	18	29.059	10
5	23561	860.78	24	867.61	8
6	23561	4.5617	15	161.82**	2
7	23561	14.19**	16	43.499*	14
8	23343	45.819	37	65.682*	4
9	23343	55.918	16	457.08	6
10	15772	15.615	18	21.522	14
11	21610	45.456	18	25.592	9
12	22830	57.773	22	178.88*	4
13	15354	40.656	22	44.951	6
14	25994	63.696	17	211.02	6

Table 8: Orbit Update Comparison from EBC and CMOS Data for Single Pass Cases. A * indicates that the convergence criterion was adjusted to 0.01, while a ** indicates that the convergence criterion was adjusted to 0.1.

Orbit updates for single observations were calculated in a similar manner to orbit updates from multiple observations, with the only difference being that the future TLE at which the comparison was made was from the day directly following the observation.



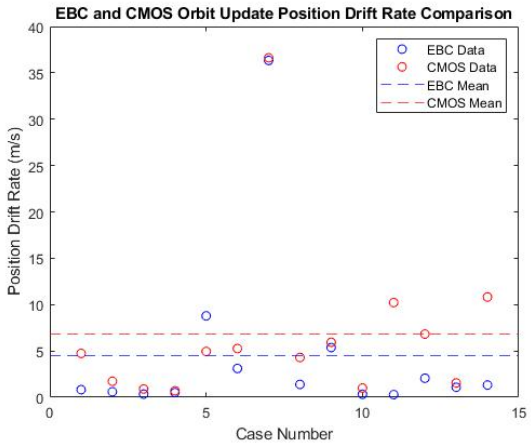
(a) Position Error Comparison.



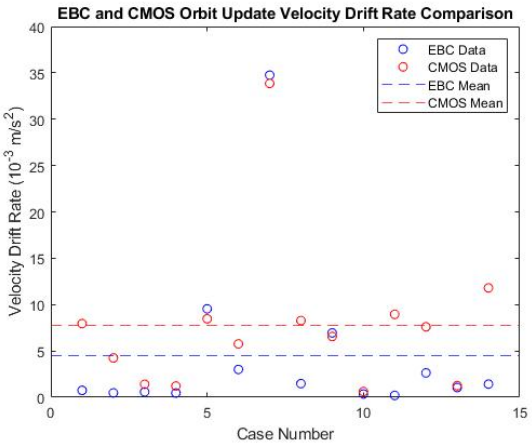
(b) Velocity Error Comparison.

Figure 31: Plot comparing the relative number of EBC data points to CMOS data points and the corresponding difference in accuracy of the orbit update in each case.

Figure 31 shows that, while the variance of the data is much larger, the fact that most data points fall above the break-even line indicates that orbit updates produced by data from the EBC were typically more accurate than those produced by data from the CMOS sensor.



(a) Position Drift Comparison.



(b) Velocity Drift Comparison.

Figure 32: Plot comparing the drift rate magnitudes of the orbit update provided by the EBC and CMOS sensor for each case.

Figure 32 indicates that, as the more accurate EBC orbit updates shown in Fig-

ure 31 would imply, the drift rates of orbits updated from a single EBC observation are lower than those for single CMOS observations. In other words, single observation orbit updates made from EBC data accumulate error at a lower rate than updates from CMOS data.

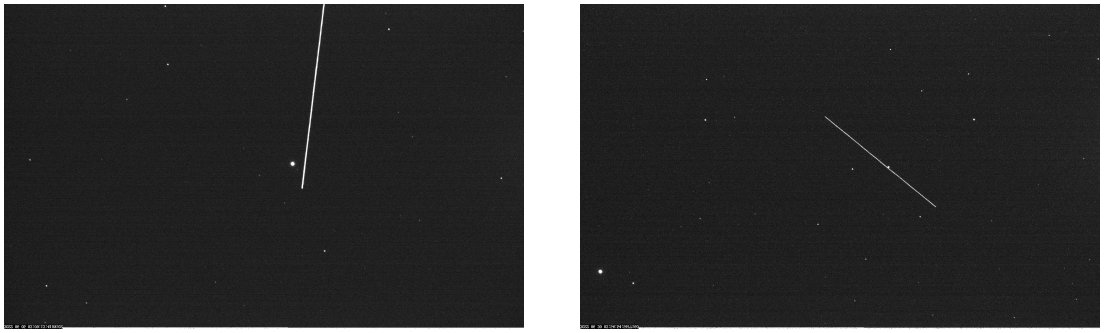
4.2 Analysis of Results

Based on the results, two takeaways become apparent: the first is that it is indeed possible to produce an orbit update from EBC data, which will be discussed in more detail in Chapter V, and the second is that the ability of an EBC to produce more data points than a CMOS sensor seems to play an important role in this.

A fundamental limitation of CMOS sensors is that, because a streak needs two ends visible in the frame to provide two data points, a streak that enters or exits the frame can only provide one certain data point in that frame. Given how few data points are produced by a CMOS sensor in optimal conditions, losing data points in this manner can significantly impact the accuracy of an orbit update. Because EBCs do not use frames to generate data, they do not lose data when a satellite enters or exits their Field of View (FOV).

This difference is what led to the outlier points in Figure 31 from single pass case 6. Data from the CMOS sensor was lost in this manner in the case where 7.5 EBC data points were produced for each CMOS data point, while cases with lower EBC to CMOS data point ratios were made possible by the RSO streaking through the diagonal of the CMOS sensor. The difference between these cases was that a satellite passing through a rectangular frame vertically travels through less of the sensor's FOV than a satellite passing through the diagonal, such that a sensor that collects data in frames of a set duration can be severely limited by the orientation of the pass with respect to the sensor's FOV due to not capturing the start or end points of a

streak.

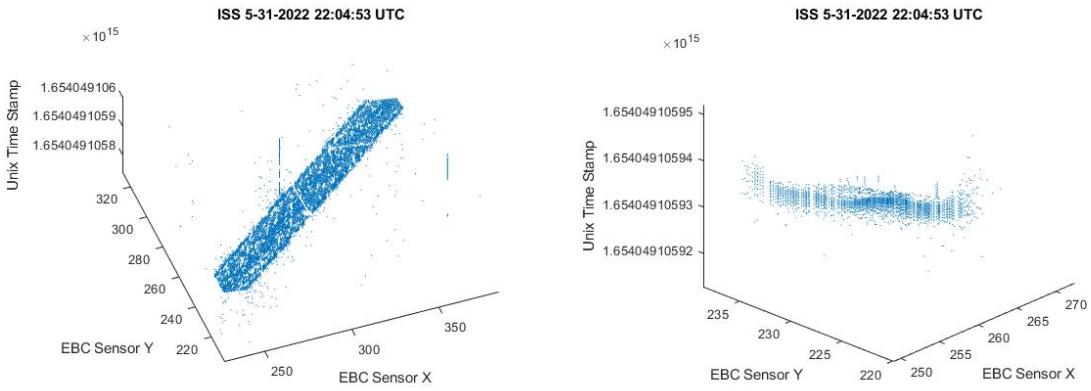


(a) Case 6 only had 2 CMOS data points. (b) This pass generated 8 CMOS data points.

Figure 33: An example of how the orientation of a pass with respect to the FOV of the CMOS sensor can impact the number of data points collected.

It is important to note that this problem can be mitigated by taking shorter exposures to generate shorter streaks, and thus more data points. However, if only the shutter speed were changed, the amount of signal per frame available to the CMOS would also be reduced which could negatively affect the plate solving process. A higher gain could be used to compensate for this, but the point is that frame-based sensors have a fundamental vulnerability to which EBCs are not susceptible.

This discussion of the number of data points generated per pass also brings up the question of why, for a system that can collect more data points than a CMOS sensor by about two orders of magnitude, the ratio of EBC to CMOS data points along the x axis in Figures 29 and 31 is not higher. The reason more EBC data points are not used to calculate each orbit update is that there is a difference between the number of data points collected per pass and the number of data points per pass that can be meaningfully processed into an orbit update, as shown in Figure 34.



(a) Top view (b) Cross section view

Figure 34: Plot of part of an ISS observation that shows how many events are generated by a bright object.

An RSO pass will not always generate a perfect line of events across the sensor of the EBC like in Figure 25. Instead, bright RSOs are capable of generating a streak about 15 pixels (34.2") wide and 10 pixels tall. Processing every single data point from a streak would provide contradictory time and position data to the orbit update algorithm, so only the data points selected as a best representation of the streak by the individual processing the data were used to generate the orbit update. Though helpful for finding streaks, the Hough transform detailed in Section 3.3.2.1 often highlighted the diagonal of a streak rather than the longitudinal axis of the streak itself and was primarily used to inform the manual selection of points along a streak.

4.3 Other Results

4.3.1 Bright Sky Observations

While conducting research, the EBC was observed producing events from RSOs in the western portion of the sky as early as 15 minutes after sunset when the sky was still too bright for the CMOS sensor to easily detect RSOs, as shown in Figure 35.

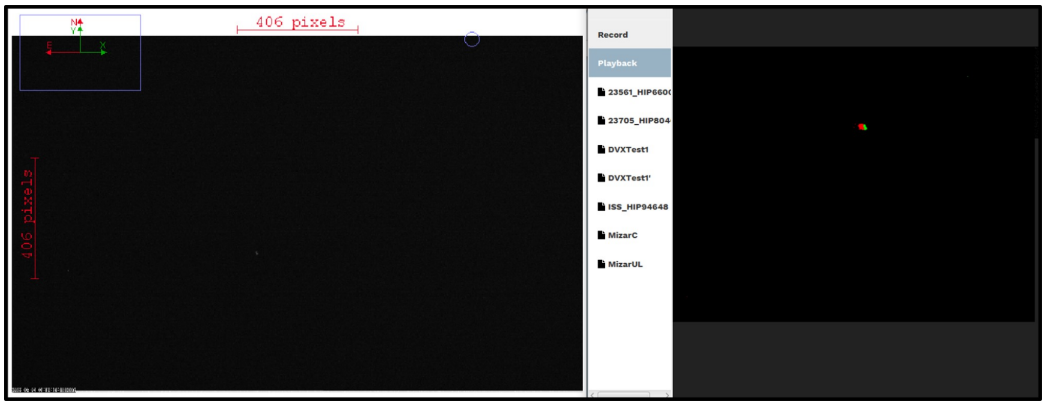


Figure 35: Screenshot showing a comparison of CMOS data (left) of the ISS with EBC data from the same pass, recorded at 21:15:10 near an azimuth of 281.84° and an elevation of 26.09° after a 21:00 sunset. The ISS is barely visible near the center of the CMOS data, but is clearly visible in the EBC data.

While daytime RSO detection has already been performed [8] and is not considered to be within the scope of this research, this data is incredibly promising for future research in the ability of an EBC to provide orbit updates from RSOs observed during the day when the brightness of the sky would otherwise prevent traditional frame-based sensors from capturing useful data. In this instance, the telescope was pointed to the west at a low elevation 15 minutes after local sunset, so the sky was still rather bright when the data was collected.

4.3.2 Noise

The primary limitation of this research was the noise generated by the EBC, which on multiple data sets prevented the successful detection of an RSO streak within a point cloud. The sensor was not designed to operate in low light conditions, so the noise floor that had to be overcome prior to data extraction added a significant burden to data processing.

The EBC sensor noise appeared as random events with no apparent stimuli, as hot pixels that would continuously produce a signal, and as dynamic pattern noise

that appeared to be the result of pixels interacting with each other. Hot pixels and random noise could be removed with some processing, but the appearance of noise that varied in intensity and position with time proved to be a significant impediment to data processing.

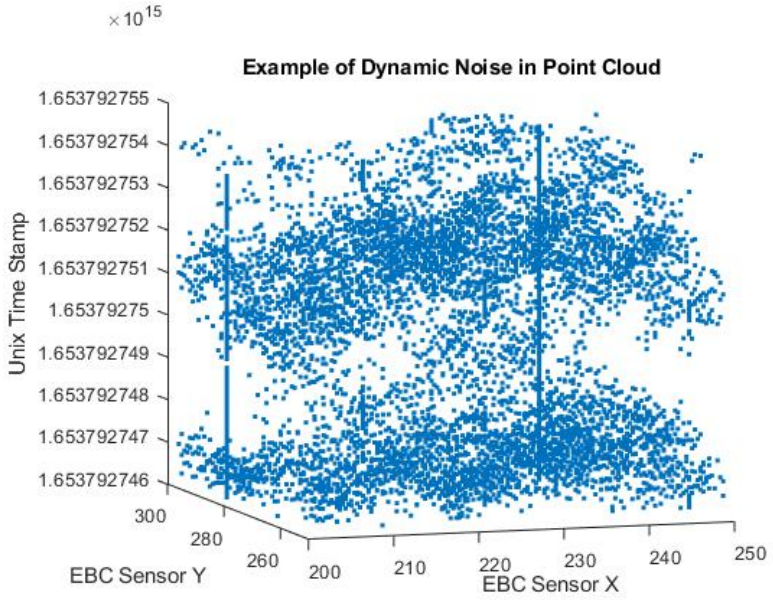


Figure 36: Example of time varying noise in an EBC data point cloud. Two hot pixels are also visible as columns of events.

The problem was that the noise pattern appeared to flow around the frame of the image in a manner that appeared to indicate that the noise events were dependent on one another. As shown in the figure above, the noise came in 'waves', and appeared to propagate around the sensor's pixel space.

The EBC used for this research could be commanded to record on a lower than default gain setting, but sending a signal to the sensor to reduce the sensitivity was observed on multiple occasions to create more waves of noise and more event counts per second rather than fewer. Attempts to use higher gain settings were met with the same problems. As a result, all data was recorded on the default gain setting so as to avoid inducing noise by sending commands to the sensor.

4.3.3 Pixel Interactions

Another quirk of the EBC sensor was that a bright enough object appeared to have the ability to activate events in parts of the frame that it had not occupied while data was being recorded, as shown in the following figures:

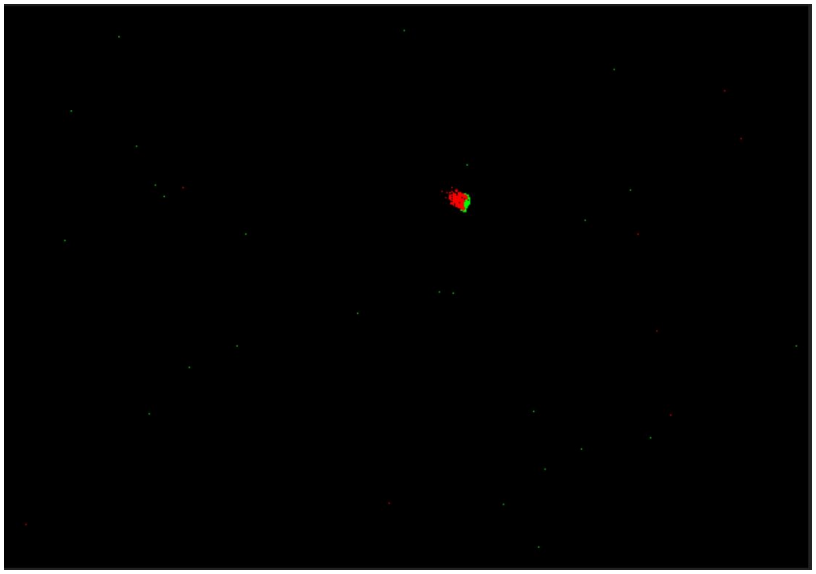


Figure 37: Data present at the beginning of the recording. Note the position of the RSO in the frame.

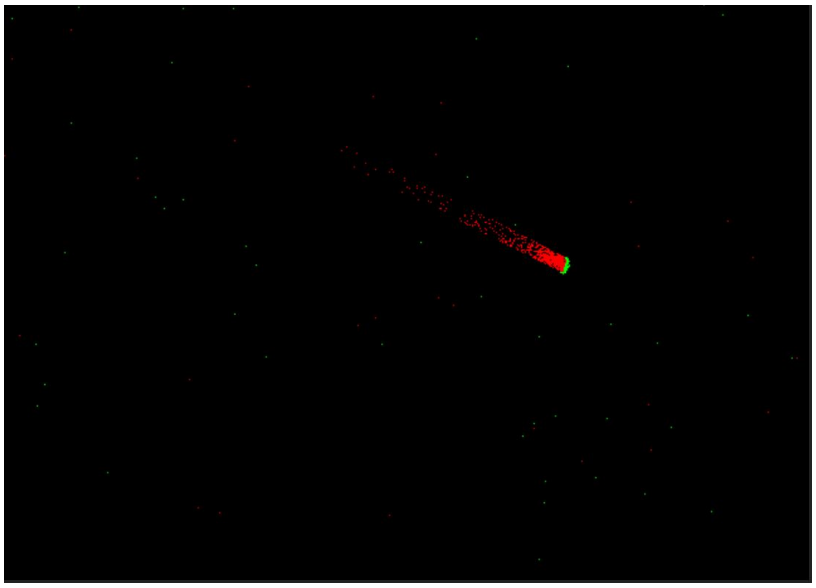


Figure 38: Example of negative events spreading behind an RSO's position. Note that the very faint red trail of negative events extends past the position of the RSO in the previous figure.

The context to these figures is that a recording of a bright satellite was started while the satellite was in the center of the frame moving towards the bottom right corner of the frame. Despite it not crossing through the top left corner of the frame during recording, negative events were observed there anyway, apparently part of a trail of events that began to grow from the satellite's position after recording began. While these extra events may prove useful for RSO detection in cases where the exact location of an RSO is not needed, the extraneous events were not beneficial in this case. The cause of this phenomenon is unknown, though it is likely due to the brightness of the RSO being observed causing pixels to leak signal into surrounding pixels. Regardless of the mechanism, this will need to be accounted for prior to any operational use of EBCs for Space Domain Awareness (SDA).

V. Conclusions

5.1 Analysis

The research question for this thesis was to determine whether or not the increased temporal resolution of a current Commercial Off-The-Shelf (COTS) Event-Based Camera (EBC) could make up for its relatively low spatial resolution when compared to frame-based sensors for the purpose of providing data for orbit determination. While EBCs operate in a fundamentally different manner than frame-based sensors that allows them to generate far more data points per unit time than a frame-based sensor, their relatively new technology has not yet achieved the spatial resolution of current COTS Complementary Metal-Oxide Semiconductor (CMOS) sensors, and their capability to serve as sensors for Space Domain Awareness (SDA) applications has not yet been quantified.

Figure 29 and Figure 31 directly answer this question by showing the relation between the number of data points produced by each sensor and the relative accuracy of the orbit updates provided by the data from each sensor. The data shows that, because the EBC produced data with a higher temporal resolution than the CMOS sensor, and because the EBC produced more data points than the CMOS sensor, the EBC has the ability to produce more accurate orbit updates than the CMOS sensor.

In other words, the EBC can certainly compensate for its lack of spatial resolution with its ability to sample data with a higher temporal resolution than a CMOS sensor, which allows the EBC to record more data points from a single pass of a Resident Space Object (RSO). Of the 19 total test cases, in 14 the EBC produced an orbit update that resulted in a future state closer to the truth data than the CMOS could, and in none of the 5 cases where the CMOS sensor produced a better orbit update was the EBC much worse than the CMOS sensor in terms of accuracy. This is made

even more impressive by the fact that the EBC's sensor size was about one third of the CMOS sensor, which means that it was still able to compete with the CMOS sensor for more accurate orbit updates despite being able to sample even less of each satellite's orbit than the CMOS sensor could.

However, it was difficult to determine how exactly the EBC compensated for its relatively lower spatial resolution. The initial concept that inspired this research was that the increased temporal resolution of an EBC could allow EBC data to generate more accurate orbit updates than a CMOS sensor, but because the limiting factor based on the analysis in Section 3.6.1 is spatial resolution, not temporal resolution, it is difficult to assert that the temporal resolution of the EBC allowed it to produce better orbit updates when the results indicate that the benefit came from the sheer number of data points per pass.

In terms of the metrics defined in Section 1.2, the orbit updates produced for this research support the assertion that current COTS EBCs can be used for SDA applications to produce results comparable to or more accurate than contemporary frame-based sensors, and that the EBC used for this research could be used for SDA applications at its current level of development, though with some difficulty due to the high dark noise levels present.

It is also important to differentiate between the COTS EBC that was used for this research and EBC technology in general. While numerous shortcomings of the EBC used here were described in Section 4.3.2, that should not be confused with EBC technology as a whole. There is certainly room for individual COTS EBCs to improve their usefulness for SDA applications, but nothing about EBC technology proved to be a fundamental impairment that would prevent EBCs from being used for SDA applications.

5.1.1 Single Pass vs. Multiple Pass

Perhaps the most interesting result to come from this research, which was not expected and not specifically asked for initially, was the relative difference in results between orbit updates that used data from multiple passes and orbit updates that produced data from a single pass.

To be clear, the best method to generate an accurate orbit update is to collect as much data as possible over multiple passes, but in some cases where the RSO of interest maneuvers or where time for observing is limited there may only be one opportunity to collect data for an orbit update. Based on the data, this suboptimal scenario of only having a single pass with which to produce an orbit update is where the EBC truly appears to have an advantage of the CMOS sensor.

Multiple pass orbit updates were generally more accurate than single pass orbit updates based on the magnitude of position error, but as shown in Figure 29 most of the EBC data points fell below the break-even line, indicating that they were less accurate than the CMOS orbit update produced from the same observations. This is to be expected, because the increased spatial resolution of the CMOS and the ability to leverage data points over a longer duration produces ideal conditions to generate an accurate orbit update from frame-based sensor data. This makes sense, because frame-based sensors that contribute data to the Space Surveillance Network (SSN) operate in a similar manner.

However, most of the data points in Figure 31 were above the break-even line, indicating that in most cases the orbit update produced by the EBC from a single pass was more accurate than the orbit update produced by CMOS data for the same observation. This is a very valuable result that indicates that not only could the EBC's ability to generate more data points compensate for its reduced spatial resolution, when the data available is limited to coming from a single pass, it can even produce

a better orbit update than a frame-based sensor.

This result has great significance for SDA applications, where the growing number of RSOs mentioned in Section 1.1 has increased the pressure on the SDA community to do more with less. If EBCs can prove to be more efficient at generating orbit update data than frame-based sensors, they would certainly be useful for SDA applications, which has been the main focus of this research.

5.1.2 Sensor Improvements

That said, there is still a clear path to improvement for EBCs to become significantly more effective for SDA than frame-based sensors. Throughout the process of collecting and analyzing data it became apparent that, rather than the fundamental mode of operation of the EBC, the limiting factor of using the EBC for orbit updates was the ability to pull the signal out of the noise. The options on the COTS EBC market are not intended for low-light data collection and could benefit from improvement in multiple areas to make them better suited for SDA applications.

The first of these areas is the low-light performance of the EBC. While EBCs are nominally efficient at producing data because they only record when they detect events, the reality is that their exceptionally high dark noise level can obscure the data from being detected in a point cloud. Data from the EBC often resembled the point cloud shown in Figure 20, where the streak generated by the satellite is visible but rather difficult to isolate. Advanced noise reduction and streak detection algorithms were successfully employed in Section 3.3.2.1, but at the expense of spending valuable computing time spent to process the data into a readable format. Whether this takes the form of improved pixel technology or by adding a thermoelectric cooler to the sensor, the amount of sensor noise makes current COTS EBCs difficult to use for SDA applications. If EBCs are to be used operationally for orbit determination

without closely supervised noise removal processes, the dark noise level will need to be significantly reduced to ensure automated processes are robust.

The next aspect of EBCs that should be improved is the size of the sensor itself. While small sensor sizes are sufficient for recently released EBCs intended for industry [35], keeping EBC technology small is not beneficial for SDA applications. At only 5.76 by 4.32 millimeters, the sensor of the EBC that was used for this research covered less than a third of the area and just over half of the width of the CMOS sensor. Because of this, the EBC could sample even less of the orbit than the CMOS sensor, which for a single observer at a single location was already sampling a minuscule section of the orbit. Given this limitation, it is truly remarkable that the EBC could produce orbit updates that could compete with those produced by the CMOS sensor. If the sensor size of the EBC were equal to that of the CMOS sensor with the same size pixels such that the Field of View (FOV) of the EBC were larger, it is likely that the orbit updates produced by the EBC would have been better than those produced by the CMOS sensor.

A related aspect of the EBC that should also be improved to better suit EBCs for SDA applications is the size of the individual pixels on the sensor. Between the sensors used for this research, the EBC's pixels were 1.5 times larger than those of the CMOS sensor, resulting in the EBC having a spatial resolution 1.5 times larger than that of the CMOS sensor. EBCs with improved spatial resolution similar to what Sony has recently released [35] are especially important because EBCs cannot take advantage of processes like centroiding that help CMOS sensors achieve sub-pixel spatial resolution [33]. Because EBC data is binary and cannot provide any light curve data, it is effectively impossible to determine which pixel (or position in a pixel) contains the center peak of a star or RSO in EBC data. In terms of SDA, this makes astrometry with existing EBCs very difficult and significantly increases

the uncertainty associated with selecting points along the streak of an RSO.

These improvements must come from the EBC itself and cannot be made by using a different telescope. If a telescope with a longer focal length were used with the current EBC the spatial resolution would improve, but at the cost of reducing the FOV of the EBC, which would allow it to observe even less of an RSO's orbit. Reducing the focal length of the telescope to increase the EBC's FOV would come at the cost of reducing spatial resolution further, which is also undesirable. The path to more effectively using EBCs for SDA is through larger sensors and smaller pixels, not through different optics.

While certain newer models of EBC have improved over the DVXplorer that was used for this research, the intended use case of newer EBC models is still in the industrial sector [35]. Until a company begins producing EBCs designed for the intended use case of observing stars and RSOs, it is likely that COTS EBCs will continue to be only a partial solution for SDA applications with ample room still left to improve. For example, producing an EBC with smaller pixels but no improvement in light gathering ability per pixel would generate data with better spatial resolution, but because the smaller pixels would have less surface area with which to absorb photons, the limiting visual magnitude of the sensor would decrease in a manner that would make the sensor poorly suited to observing dim objects. Improving the spatial resolution while maintaining the same sensor dimensions also does not address the issue of smaller sensors having a smaller FOV that limits their ability to observe more of a satellite's orbit. The existing COTS EBC option used for this research produced orbit updates for bright Low Earth Orbit (LEO) objects but still had trouble capturing data on even some dimmer LEO objects, so attempting to improve EBCs by trading one aspect of performance for another will likely lead to overall EBC performance falling below the minimum required to observe anything but the brightest

RSOs.

Until solutions exist that address tradeoffs such as the ones exemplified above, it is unlikely that EBCs will surpass traditional frame-based sensors by a margin wide enough to justify their operational use. This thesis has shown that EBCs can be used to produce orbit updates comparable to those produced by a CMOS sensor when provided with plate solutions from the CMOS sensor, but further work will need to be done to demonstrate that EBCs alone are well suited for operational SDA applications.

5.2 Future Work

Throughout the process of this research it became apparent that there were multiple other avenues of study that would yield valuable research results, but would either require too much precious time to be included in this thesis or require EBC technology better than the current COTS options.

- Now that the ability to update an orbit with EBC data has been established, more research should be done to investigate whether the advantage comes from the number of data points, the temporal resolution of each data point, or a combination of both. This research only demonstrated that it could be done and that the number of data points seems to be more important, but until an experiment that controls for the number of data points is performed, it will be difficult to definitively say what allows an EBC to produce better orbit updates than a CMOS sensor.

- More work should be done to automate EBC streak processing and the conversion of data from the point cloud to data points in the EBC pixel space with corresponding time stamps. This could be performed using the Hough

transform [30], which has already been used on EBC data to some success [36]. While it is currently possible to generate a model of the streak with the Hough transform, the ideal end state would be to use the Hough transform to find all of the EBC data points nearest the best fit line and select those for processing.

Future students could also experiment with noise reduction techniques such as refractory period or nearest neighbor filters. These have already been applied to EBC data intended for object tracking with varying degrees of success [9], and because they do not rely on lines in data the way a Hough transform does, they may prove to be more generally applicable than the Hough transform.

- The work done in this thesis should be revisited as industry develops better EBCs and AFIT acquires better optical telescope hardware. One of the primary limitations of the EBC was its sensor noise, so with a lower noise floor from improved sensor technology and increased signal from a telescope with a better limiting visual magnitude than the one used here it is likely that a similar experimental setup could be used to track dimmer objects in Medium Earth Orbit (MEO) and Geosynchronous Equatorial Orbit (GEO). GEO observations with EBCs have already been made [8], but not with a beamsplitter sharing data between a CMOS sensor and an EBC.

- In tandem with research into dimmer objects, a student equipped with a better sensor and a better telescope could also research the ability of an EBC to detect enough stars to produce a plate solution. Something similar has already been done outside of the AFIT community [20], but with certain caveats that could be overcome with better equipment. Plate solutions with only EBC data would remove the need to split light with a CMOS sensor, which would improve the signal available to the EBC for additional future work.

- A student unenthusiastic about spending late nights with a telescope and some cameras could investigate the ability of an EBC to detect satellites during the daytime and produce orbit updates based on the data using methods similar to those described here. Daytime satellite detection has also already been done outside of the AFIT community [8], but orbit updates have not yet been made from EBC data recorded during the day.

5.3 Real-World Implications

While most of this thesis has been approached from an academic perspective, it would be a failure on my part to omit the real-world applications of this work.

Our next conflict will almost certainly involve kinetic effects directed towards space-based assets at the outset as both sides attempt to disable each other's space-based support systems. To win, it is absolutely imperative that we maintain SDA so that we already know where to find the assets that we want to defeat, and how close they are to assets we want to protect. To do this, we cannot solely rely on frame-based sensors that were developed and perfected in an era when we could afford the time to observe an asset for a few passes to determine its orbit. In a modern space conflict, we will almost certainly be denied that luxury, and will instead need to focus on sensors that can provide as much data in a single pass as possible to track assets both prior to and directly after maneuvers. For this purpose, EBCs should have a place in our inventory of SDA sensors due to their ability to provide many data points during a single pass and to observe objects during the day. While this research is preliminary, there is now a clear path to operational use that must be followed if we want to fully take advantage of EBCs for SDA.

Bibliography

1. Daniel Moomey. A call to action: Aid geostationary space situational awareness with commerical telescopes, 2015.

2. Johnson Space Center. Orbital debris quarterly news, 4 2009.

3. Johnson Nicholas. The collision of iridium 33 and cosmos 2251: The shape of things to come, 8 2009.

4. John Raymond. Space capstone publication. 2020.

5. Teledyne DALSA. Ccd vs. cmos.

6. Tobi Delbruck. Neuromorophic vision sensing and processing. volume 2016-October, pages 7–14. Editions Frontieres, 10 2016.

7. Joseph G Bacon. Satellite tracking with neuromorphic cameras for space domain awareness, 3 2021.

8. Gregory Cohen, Saeed Afshar, André Van Schaik, Andrew Wabnitz, Travis Bessell, and Mark Rutten. Event-based sensing for space situational awareness, 2017.

9. James P Boettiger. A comparative evaluation of the detection and tracking capability between novel event-based and conventional frame-based sensors air force institute of technology, 3 2020.

10. Stefano Meroli. Active pixel sensor vs ccd. who is the clear winner?, 4 2012.

11. Misha Mahowald. *An Analog VLSI System for Stereoscopic Vision*. Springer US, 1994.

12. Ryad Benosman, Sio Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 3 2012.

13. David Vallado. *Fundamentals of Astrodynamics and Applications*. 4th edition, 2013.

14. NOAA. Celestial sphere illustration.

15. American Institute of Physics. Follow the drinking gourd ra dec handout.

16. Daniel Moomey. Aiding geostationary space situational awareness using small aperture commercial telescopes, 2015.

17. Sam Roweis, Dustin Lang, Keir Mierle, David Hogg, and Michael Blanton. Making the sky searchable: Fast geometric hashing for automated astrometry, 2006.

18. 18th Space Defense Squadron. 18th sds tweet, 2 2018.

19. Jessica Horn. Neuromorphic vision sensors for space-based applications, 2021.

20. Gregory Cohen, Saeed Afshar, and André Van Schaik. Approaches for astrometry using event-based sensors, 2018.

21. Takahashi Seisakusho Ltd. Fsq-106ed instruction manual.

22. Thorlabs. Non-polarizing cube beamsplitters (400 - 700 nm), 2022.

23. Audra Jensen, Michael Plummer, Daniel O'Keefe, and Francis Chun. Observations of satellites using near-simultaneous polarization measurements, 2021.

24. QHYCCD. Qhy174gps.

25. iniVation. Specifications-current models, 8 2021.

26. Karen A. Collins, John F. Kielkopf, Keivan G. Stassun, and Frederic V. Hessman. Astroimagej: Image processing and photometric extraction for ultra-precise astronomical light curves. *The Astronomical Journal*, 153:77, 1 2017.

27. R Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Event probability mask (epm) and event denoising convolutional neuralnetwork (ednncnn) for neuromorphic cameras. *arXiv preprint arXiv:2003.08282*, 2020.

28. MATLAB. *version 9.10.0.1739362 (R2021a)*. The MathWorks Inc., 2021.

29. Eric Jensen. How can i replicate astroimagej's pixel to ra/dec algorithm in my own code?, 7 2022.

30. Christoph Dalitz, Tilman Schramke, and Manuel Jeltsch. Iterative hough transform for line detection in 3d point clouds. *Image Processing On Line*, 7:184–196, 2017.

31. William E. Wiesel. *Modern Orbit Determination*. 2 edition, 2010.

32. Neil Thomas. Astroengr 321 fall 19 coursebook, 2019.

33. Xiaowei Wan, Gangyi Wang, Xinguo Wei, Jian Li, and Guangjun Zhang. Star centroiding based on fast gaussian fitting for star sensors. *Sensors*, 18, 8 2018.

34. iniVation. Understanding the performance of neuromorphic event-based vision sensors, 5 2020.

35. Sony Semiconductor Solutions Corporation. Sony to release two types of stacked event-based vision sensors with the industry's smallest*1 4.86m pixel size for detecting subject changes only delivering high-speed, high-precision data acquisition to improve industrial equipment productivity, 9 2021.

36. Saeed Afshar, Andrew P Nicholson, Andre van Schaik, and Gregory Cohen. Event-based object detection and tracking for space situational awareness. 11 2019.

Acronyms

AIJ AstroImageJ. 37, 42

AMOS Advanced Maui Optical and Space Surveillance Technologies Conference. 23, 24

CCD Charged Coupled Device. 9, 23

CMOS Complementary Metal-Oxide Semiconductor. 6, 7, 9, 11, 12, 13, 22, 24, 25, 27, 28, 30, 31, 32, 33, 34, 37, 39, 40, 41, 42, 43, 49, 50, 51, 52, 60, 61, 62, 63, 64, 66, 67, 69, 70, 71, 72, 73, 77, 78, 79, 81, 83, 84

COE Classical Orbital Elements. 55, 56

COTS Commercial Off-The-Shelf. iv, 2, 3, 5, 6, 8, 23, 33, 34, 77, 78, 80, 82, 83

EBC Event-Based Camera. iv, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 35, 38, 39, 40, 41, 43, 44, 45, 46, 49, 50, 51, 52, 60, 61, 62, 63, 64, 66, 67, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85

ECI Earth-Centered Inertial. 58

FITS Flexible Image Transport System. 30, 37, 38, 42, 43, 49, 51

FOV Field of View. 10, 12, 16, 24, 25, 33, 40, 62, 70, 71, 81, 82

GEO Geosynchronous Equatorial Orbit. 23, 24, 84

GPS Global Positioning System. 32, 34, 51, 63

GST Greenwich Sidereal Time. 58

IDE Integrated Development Environment. 38

LEO Low Earth Orbit. 23, 24, 82

MEO Medium Earth Orbit. 84

RSO Resident Space Object. 1, 2, 4, 7, 24, 36, 37, 38, 40, 41, 42, 43, 44, 45, 46, 49, 55, 57, 58, 59, 60, 61, 62, 63, 67, 70, 72, 73, 76, 77, 79, 80, 81, 82, 83

SDA Space Domain Awareness. iv, 2, 3, 4, 5, 6, 8, 12, 22, 23, 25, 76, 77, 78, 80, 81, 82, 83, 85

SEZ South-East-Zenith. 59

SIP Simple Imaging Polynomial. 18, 25, 30, 37, 38, 42, 43, 49

SSN Space Surveillance Network. 20, 21, 60, 64, 79

SWaP Size, Weight, and Power. 22

TLE Two-Line Element. 7, 21, 39, 40, 52, 53, 55, 56, 60, 64, 65, 67, 68

USSF United States Space Force. 1, 2, 20

WCS World Coordinate System. 17, 25, 37, 38, 42, 43, 49