Sohbrit: Autonomous COTS System for Satellite Characterization

Nicholas Blazier, Samuel Tarin, Mitchell Wells, Nathanael Brown, Prabal Nandy, Drew Woodbury
Sandia National Laboratories

ABSTRACT

As technology continues to improve, driving down the cost of commercial astronomical products while increasing their capabilities, manpower to run observations has become the limiting factor in acquiring continuous and repeatable space situational awareness data. Sandia National Laboratories set out to automate a testbed comprised entirely of commercial off-the-shelf (COTS) hardware for space object characterization (SOC) focusing on satellites in geosynchronous orbit. Using an entirely autonomous system allows collection parameters such as target illumination and nightly overlap to be accounted for habitually; this enables repeatable development of target light curves to establish patterns of life in a variety of spectral bands.

The system, known as Sohbrit, is responsible for autonomously creating an optimized schedule, checking the weather, opening the observatory dome, aligning and focusing the telescope, executing the schedule by slewing to each target and imaging it in a number of spectral bands (e.g., B, V, R, I, wide-open) via a filter wheel, closing the dome at the end of observations, processing the data, and storing/disseminating the data for exploitation via the web. Sohbrit must handle various situations such as weather outages and focus changes due to temperature shifts and optical seeing variations without human interaction. Sohbrit can collect large volumes of data nightly due to its high level of automation. To store and disseminate these large quantities of data, we utilize a cloud-based big data architecture called Firebird, which exposes the data out to the community for use by developers and analysts. Sohbrit is the first COTS system we are aware of to automate the full process of multispectral geosynchronous characterization from scheduling all the way to processed, disseminated data. In this paper we will discuss design decisions, issues encountered and overcome during implementation, and show results produced by Sohbrit.

1. INTRODUCTION

As part of Sandia National Laboratories' growing endeavor to establish a baseline of Space Situational Awareness (SSA) capabilities for electro-optical (EO) sensors, an autonomous ground-based testbed has been built on-site in Albuquerque, NM. Since the cost of commercial astronomical hardware products continues to decrease while the capabilities of these products continues to increase, an automated collection and processing system was envisioned to limit the manpower necessary for the collection and acquisition of SSA data. The initial focus of this effort was to collect multispectral space object characterization (SOC) data on geosynchronous (GEO) satellites with the purpose of identifying behavioral trends and data that can be used for "fingerprinting" of the observed objects. The developed system, known as Sohbrit (SO-bright), represents the initial realization of these objectives.

The current system is capable of building an optimized collection schedule based upon user-defined target priorities and then operates the entire facility (dome, weather sensor, mount, telescope, filter wheel, camera) to perform the desired collections. Although the architecture is realized using the current hardware and software packages, the approaches and methodologies being developed for the system are intended to be sensor and hardware agnostic for applications on a large variety of EO sensing systems. Furthermore, the Sohbrit architecture is designed in a modular fashion enabling it to be a testbed for both software and hardware SSA capabilities. The following section gives a high level view of this architecture and how the different components communicate.

2. SYSTEM ARCHITECTURE

Sohbrit is composed of many components that work together to achieve automation. A major goal was to use not only commercial off the shelf (COTS) hardware but to also leverage COTS software. The major challenge in automating a collection facility comprised of COTS software is interfacing them together. Most COTS hardware provide a software development kit (SDK), or code that directly controls the hardware. This enables easier automation of the equipment, but often limits the user to given compiled languages. The Sohbrit codebase is developed in Java but many SDKs only support native programming languages such as C and C++. In order to use these SDKs in Java they must first be converted using a tool such as Java Native Access [1] (JNA). Some commercial hardware and software now offer Representational State Transfer [2] (REST) interfaces. REST interfaces allow communication through Hypertext Transfer Protocol (HTTP). This allows versatility in

programming languages and hosting platforms as all modern languages and computers support the HTTP protocol. Software that can be automated through a REST interface has the added benefit of load balancing and distribution. Since any networked computers can communicate via HTTP, applications need not be run on the same machine and can be scaled to multiple machines if necessary.

Another useful interface for automating astronomy hardware is to use an Astronomy Common Object Model (ASCOM) [3] driver. ASCOM is a common interface for all astronomy devices. Interfaces exist for devices such as telescope mounts, cameras, domes, focusers, filter wheels, and other common astronomy hardware. If implemented by the manufacturer, then ASCOM commands can be used instead of device specific SDKs or manufacturer's drivers. This allows much easier plug and play as the commands remain the same when swapping between two cameras and all that needs to be changed is the name of the camera driver. It is important to note that not all manufacturers support ASCOM drivers. Even when listed as supported, it is not uncommon to find the manufacturers ASCOM driver to be incomplete of all the features ASCOM allows, leaving advanced users to rely on the provided SDK for full hardware capability.

At a macro level Sohbrit can be broken into the scheduling, image collection, storage and retrieval, and processing subsections. Each of these sections may be further divided and are described in detail below. Often there are multiple possible solutions to a problem, when appropriate we will address decisions made and factors influencing that decision.

3. SCHEDULING

The scheduler uses a heuristic optimization to create a collection of tracked periods from the available in-view periods of the target satellites using the following multi-component objective (which is a weighted sum of the values below):

1. Minimize the amount of time for which the satellite is in view but untracked
2. Minimize the number of satellites which are completely untracked
3. Minimize the number of in-view time blocks (contiguous periods where the satellite is in view) that are completely untracked
4. Minimize (100 – minimum tracked percentage) across all satellites
5. Minimize (100 – telescope percentage utilization) over the time horizon of interest

The in-view periods are generated for each satellite over the time horizon based on propagation of the two-line element set (TLE) and location of the ground station. The periods are further processed subject to the following constraints:

1. If specified, will exclude times where a satellite falls into 'umbra' (full eclipse) or 'penumbra' (partial or full eclipse)
2. If specified, will exclude times where the solar phase angle (aka STOA or sun-target-observer angle) is greater than a certain value

Additionally, support has been built in which converts Washington double-star data (using right ascension and declination) to periods which are in-view with respect to the ground station. This procedure allows double-stars to be tracked in the same manner as satellites.

The scheduler algorithm borrows the following concepts from Genetic Algorithms (GAs) to perform the optimization in an "evolutionary" fashion:

1. Creates a large initial population to establish a collection of baseline solutions
2. Evolves the population towards better solutions

The initial population is created using both random generation as well as a greedy algorithm which attempts to build schedules that minimize the various components in the objective while still being feasible. The optimizer then executes a specified number of iterations in which a randomly selected fraction of the population is selected for improvement using heuristic techniques which "tune" each solution by driving down the objective value. The output schedule is then derived from the solution with the smallest objective and is saved as JSON-formatted text which gives the start and stop tracking times for each target (satellite or double star) as well as the azimuth and elevation at the start time. The scheduler is accessed through a REST endpoint, allowing it to be hosted in the cloud, away from the hardware that actually utilizes it.

4. IMAGE COLLECTION

The image collection subsystem is responsible for commanding the hardware to execute the schedule. A combination of software and sensors replaces the nightly setup that a human would usually perform. This system can be broken into the following tasks: focus adjustment, telescope alignment, target acquisition, and imaging.

4.1 FOCUS ADJUSTMENT

One of the major challenges of an automated system is adjusting to changing conditions. Ground based telescopes are subject to quickly changing weather conditions. Rain is an obvious hazard that must be mitigated, but there are other factors that must be taken into account in order to maximize data quality. Changes in atmospheric conditions such as temperature affect the focus of the system; Sohbrit operates in a mountain desert climate where temperatures fluctuate around 30 degrees from the hottest to coldest part of the day [4]. For this reason, it is very important to be able to adjust the focus of the system whenever conditions are deemed to have changed a significant amount. Jerry Lodriguss suggests that changes as little as 1° Fahrenheit can bring a telescope system out of focus [5]. When the focal path is changed by introducing a filter, the system must readjust focus or risk out of focus data.

Focus Max [6] is a software tool written by Larry Weber and Steve Brady for focusing telescope systems. The tool requires the user to have an automated focuser (a focuser that can move focus in and out automatically with a motor) and a camera to observe the changes. The software initially builds a V curve of the Half Flux Diameter versus the precision focus position to understand the focus of the system. Using this curve and measurements from captured images, the software can quickly bring the telescope into focus. The current version of Focus Max (v4) provides an ASCOM driver, which allows for easier automation. It should be noted that Focus Max cannot be used by itself. Focus Max does not have any camera support, meaning it must rely on other software for camera control. This requires the user to buy additional camera control software in order to use Focus Max for focusing.

4.2 TELESCOPE ALIGNMENT

The system must remain precisely aligned in order to capture targets in a small Field of View (FoV). The initial polar alignment, the act of aligning the telescope's orientation with the celestial pole, was performed by hand. However, nightly star alignment for the computerized system is an automated process. There are many commercial tools available for star alignment of a telescope. Hardware solutions exist for some telescope mounts such as the StarSense AutoAligner for Celestron mounts. The Sohbrit system favors software solutions as they are more generic across telescope manufacturers. Software techniques require a computerized mount capable of accepting pointing information, a camera to take images through, and software capable of plate solving. Plate solving is the act of pattern matching stars found in an image against known star catalogs to determine where in space the image depicts. While most computerized mounts do a good job of "remembering" where they are pointing, plate solving allows a "lost" telescope to quickly reacquire its pointing position by taking as little as one image of the night sky.

There is no shortage of software for plate solving and aligning a telescope. Many tools leverage an open source tool called Astrometry.net [7] as a backend. These tools provide a nice framework that will take an image with your camera, plate solve using Astrometry.net, and then sync the results to your telescope thus aligning your telescope with "one" action. Sohbrit currently uses Sequence Generator Pro (SGP) [8] for this task. In its application SGP works exactly as described above, but for an autonomous system there is no one to click the graphical button to accomplish this. SGP provides a REST interface capable of capturing and saving an image. It has another REST endpoint for plate solving against any image. This endpoint allows users to blind solve (the telescope is assumed to be "lost" and has no guess as to where it is) or use a refined plate solve that requires the mount to have a good general idea of where it is pointing. Sohbrit uses the blind solve at the beginning of the night to determine its initial position and then uses the refined solve before beginning each targeted collection to ensure optimal pointing accuracy. These REST endpoints return the Right Ascension and Declination of the image passed in. That information is then given to the mount through its ASCOM interface. It is important to note coordinate frames received from the plate solver and expected from the mount at this step and to ensure they match. Using a German Equatorial mount we find it advantageous to perform another solve anytime we incur a meridian flip. This becomes less necessary if your system has very good polar alignment and high-precision hardware.

4.3 TARGET ACQUISITION

Once the mount is aligned, the system can execute the schedule by slewing to each target and imaging it for the scheduled duration. We find the ASCOM interface for telescope mounts to work well. It allows for "go to"

commands in either Right Ascension / Declination (RA/DEC) or in azimuth and elevation (Az/El) relative to the telescope's location. RA / DEC pointing is well suited for astrophotography as stars keep a constant RA / DEC and sidereal tracking keeps the mount pointing at that RA / DEC for the duration of the track. Sohbrit utilizes Az/El pointing for geosynchronous targets which keep a constant Az/El for short periods of time (this time depends on the size of your aperture and imaging hardware). When imaging with relatively short exposure lengths the mount can "track" geosynchronous targets by turning its tracking motors off.

After slewing to the target we perform a second, non-blind solve which helps ensure pointing precision and usually results in well centered targets. This step is only performed if there is adequate time before the collection is scheduled to begin; otherwise the original alignment is maintained. With adequate time this would also be another opportunity to adjust focus, particularly if the weather has changed drastically since the last focus.

4.4 IMAGING

There are many ways to automate the imaging of cameras for astronomy. Many of the big astrophotography applications such as Sequence Generator Pro provide interfaces to automatically capture images. As discussed above a common interface is an HTTP-based REST interface. Using these 3rd party applications is convenient but introduces a non-deterministic timing delay. When capturing images such as Fig. 1 where a satellite is quickly moving through the frame this unknown delay is unacceptable. It is for this reason that we prefer to control any imaging hardware directly through drivers or provided SDKs.

Fig. 1. Enhanced image of COSMOS 2242 streaking through star field.

For geosynchronous characterization we run a sequence of collections on each target for the duration of the collection. Once the target is acquired, the sequence begins. From the start time of the collect until the schedule end time, the camera and filter wheel work together to characterize the target in multiple spectral bands. The filter wheel is automated through ASCOM drivers and continuously switches between U, B, V, Rc, Ic, and wide open filters in synchrony with the imaging camera. Each image is then automatically passed on to the storage and retrieval component. The result is a collection of images in each band over time that can be processed into a graph characterizing the satellites spectral features.

5. SAFETY

Humans are very good at reacting and responding to changing conditions. Without a human in the loop, a number of extra sensors are required to ensure the safety of the system. Even the most robust hardware and software will occasionally fail or act erroneously; Sohbrit is designed to mitigate as many of these potential failures as possible. Of utmost importance is ensuring that the system does not operate in dangerous conditions and cannot destroy itself due to software or communications errors.

Keeping a telescope correctly aligned is a delicate process. Plate solving to gain alignment requires correct information on the current time, location of the telescope, and good images to solve against. If any of these go wrong, it is possible for the telescope to become lost. This may seem unlikely, but consider cloudy conditions where there are not many stars to solve against, an out of focus image that confuses the plate solving algorithm, and a daylight savings time change. These events combine to greatly increase the odds of the system solving against the wrong celestial coordinates, or syncing with the telescope at the wrong time, both result in a telescope that is not aligned correctly. With no human at the site to watch the telescope, there is now an increased risk that the telescope can slew into an improper position and crash itself into the pier.

One simple mitigation strategy is to loosen the locks on the mount's motors. If set correctly the motor can still slew the mount in normal conditions, but when against pressure, such as the telescope pushing into the pier, the motors will slip and not slew the telescope any further into the pier. This should avoid any catastrophic damage to equipment, but is still unsettling.

A more robust solution is to use a limit switch. A limit switch can be used to cut the power to the motor anytime the mount moves outside of the limits set by the user. Sohbrit utilizes an angle-based limit switch controlled by an Arduino. The Arduino monitors an accelerometer mounted on the Right-Ascension axis of the telescope. Any time the mount moves beyond the user defined angles (indicating it will soon crash into the mount), power is cut to the mount and movement stops. This is used as a last line of defense against crashing into the pier. Currently the system must be reset by a human if the limit switch is triggered.

Another important safety concern that must be addressed is changing weather conditions. Albuquerque experiences monsoon season in the late summer where conditions can quickly change from sunny skies to a torrential downpour. Sudden rain spouts can damage sensitive electrical equipment such as the telescope mounts motor or the delicate cameras. Snow and hail present similar devastating risks. Sohbrit uses an AAG CloudWatcher weather station to detect dangerous conditions such as rain or high winds. Once detected the dome is automatically closed until conditions improve.

Power outages present an additional risk to the hardware. In the event that power is lost while the system is collecting, the hardware is exposed with no way of protecting itself from changing weather conditions. To alleviate this issue, an uninterruptable power supply (UPS) was installed as a backup power supply. The UPS will activate when power is lost and provide enough power to signal and subsequently close the dome to protect the hardware until power can be restored.

6. IMAGE STORAGE AND RETRIEVAL

With the Sohbrit system presenting an automated collection method for high-resolution images, the volume of data collected can quickly compound. Likewise, the automatic scheduler and filter adjustment can make searching through even a single night's worth of images difficult. Presented with these challenges, it was necessary to integrate Firebird, a cloud-based big data architecture developed at Sandia National Laboratories, into the system.

6.1 STORAGE

Firebird is designed to utilize and run in industry-standard cloud platforms including Open Stack, Amazon Web Services, and similar platforms for cloud-based computing. These technologies allow systems like Firebird to run on virtual computing resources in the cloud rather than going through the effort of maintaining physical servers. Cloud-based solutions are the industry standard for big data systems since they can scale seamlessly to meet demand by simply expanding the virtual resources in the cloud. Likewise, systems in the cloud are portable and can be deployed to numerous locations based on the specific use case. At the scale of data that the Sohbrit system is able to produce, it is very worthwhile to utilize a big data architecture that is designed for the cloud.

Some additional features of the Firebird architecture that make the system work well with Sohbrit are the shape validation spec, Kafka commit log, flexible NoSQL database, and object store. Firebird has custom specifications for the data written in a library called clojure.spec that allows validation of the data that comes into the system, not just on types (i.e. ensuring that numerical fields are in fact numbers) but also on values by ensuring for example that longitudes are between -180 and 180 degrees. The power of clojure.spec is also customizable such that it can be validated only when necessary to avoid slowdowns for optimal performance and allow flexible handling of the data. The Apache Kafka-distributed commit log is also another key component. It allows for the data to be entered into the system in a persistent stream that can be read by many processes at once to eliminate data loss within the system and enable stream processing on the data as it comes in. An open-source NoSQL database called MongoDB is the main database that is used within Firebird. MongoDB offers some advantages over traditional databases such as horizontal scalability optimized for the cloud and streamlined development requiring minimal setup for storing data. Lastly, Firebird implements an object store solution for raw files which allows storage of images separate from the metadata that is optimized for searching in the database. This allows for easy access to the files while the information describing the files is stored away safely in the database. Together these features make the Firebird architecture well suited for storing data that Sohbrit produces.

6.2 RETRIEVAL

Sohbrit data must be easily accessible and extractable from a system that can automatically produce large amounts of data. This data is produced in an order decided by scheduling, not initially optimized for processing. Retrieval of the data may be required on any number of criteria, so an optimized search needs to work reliably with many different types of queries. As with storage, this issue is solved with the Firebird data architecture system.

Firebird solves this problem using the GraphQL[9] query language, which was created internally at Facebook, but has gained widespread adoption and development in the open-source community. GraphQL is a query language that can be used by an application to gain comprehensive data retrieval capabilities, including data ranges, matching, search, and filter. At the same time, GraphQL abstracts away many complexities, allowing the user to avoid worrying about the details of the underlying database, network traffic, or synchronizing the client and the server. GraphQL is a powerful capability layer that Firebird utilizes to allow for in-depth queries across the entire Firebird system.

6.3 MULTI-PHENOMONOLOGY DATA

The Firebird architecture is already being used for multiple other types of data. This provides the opportunity to retrieve both the EO data from Sohbrit as well as other sources from differing phenomenologies to obtain higher fidelity results. Furthermore, the combination of different sources of data can even create new conclusions not considered before the data sources were combined in the same system. These design decisions and their usage in the Sohbrit project have opened new and future avenues for exploration of data across many disciplines.

7. PROCESSING

Though the focus of automation for Sohbrit is on the collection component, we have also done work on automating the processing of the data. We leverage an internal Python/MATLAB-based nodal pipeline architecture for processing. This allows us to easily swap in and out components as we develop new algorithms or analyze performance. A pipeline-based architecture allows for easy automation and flexible use. Since processing occurs when the data is fed into the pipeline, data demands can dictate whether the raw data is fed in immediately after collection, or stored to be processed at a later date. This structure allows flexibility for machines that have limited processing support. Important collections can be immediately processed and available for retrieval while less desired collects can delay processing until they are requested by a user.

Sohbrit is currently focused on geosynchronous satellite characterization and uses this pipeline setup to process information from raw images into informative plots. After the hardware collects a number of images across each of the filters listed in Section 9, a pipeline is run to detect all satellites in the field of view, analyze their spectral characteristics, correlate targets across images into tracks, and finally plot the spectral characteristics over time. There is no shortage of tools or algorithms to accomplish each of these tasks, the utility of our pipeline allows us to easily switch algorithms if they are not meeting performance goals, or for algorithm comparison and evaluation.

Fig. 2. Raw Image of NORAD ID 27378 (middle) and two other unknown GEO targets with stars streaking through image.

Astrometry.net [7] is an open-source tool that quickly analyzes any celestial image to determine various information such as the celestial coordinates, field of view measurements, and stars detected. Since our images are of geostationary targets with no tracking from the mount, we have images with streaked stars and point sources for targets. In an analysis of the capability of Astrometry.net's star detector, we use it as the first step in our pipeline to detect satellites (which appear similar to stars from a mount tracking at the sidereal rate). Astrometry.net also provides the spectral intensities which we later use to construct plots over time.

Once targets are detected in all of the images, there is an abundance of algorithms we can use to correlate them into tracks. Again we can leverage the pipeline architecture to compare and contrast various algorithms. We currently use an in-house algorithm developed as a summer intern project which implements an auction algorithm. Auction algorithms, originally proposed by Dimitri Bertsekas [10], are distributed algorithms designed to solve the assignment problem, or finding optimal pairings between a bi-partite graph. In this case the nodes on the graph are the detected targets in each of two subsequent frames and the edges between the nodes represent the pixel distance between the pairs of targets. The algorithm finds the pairings between the two frames that optimizes the graph. This establishes a track between the two frames. The process is repeated for subsequent frames, resulting in satellite locations tracked and correlated across the entire collection of image.

Now that we have the targets correlated across the entire dataset along with spectral information, we can create plots over time. Fig. 2 shows one raw image pulled from the collection. Fig. 3 shows the resulting graphs for the 3 targets captured in the field of view as Peak Pixel Wellfill (peak percentage of pixel saturation, where 1.0 means the pixel fully saturated) versus Frame Number. The targets are numbered from right to left as seen in Fig. 2. Target 2, the middle target, has saturated the camera in a number of frames indicating a need for better calibration. The automation of the scheduling, collection, and processing makes it much easier to perform repeat collections. This will allow for the vast quantity of data necessary for analysists or machine learning algorithms to spot patterns and anomalies.

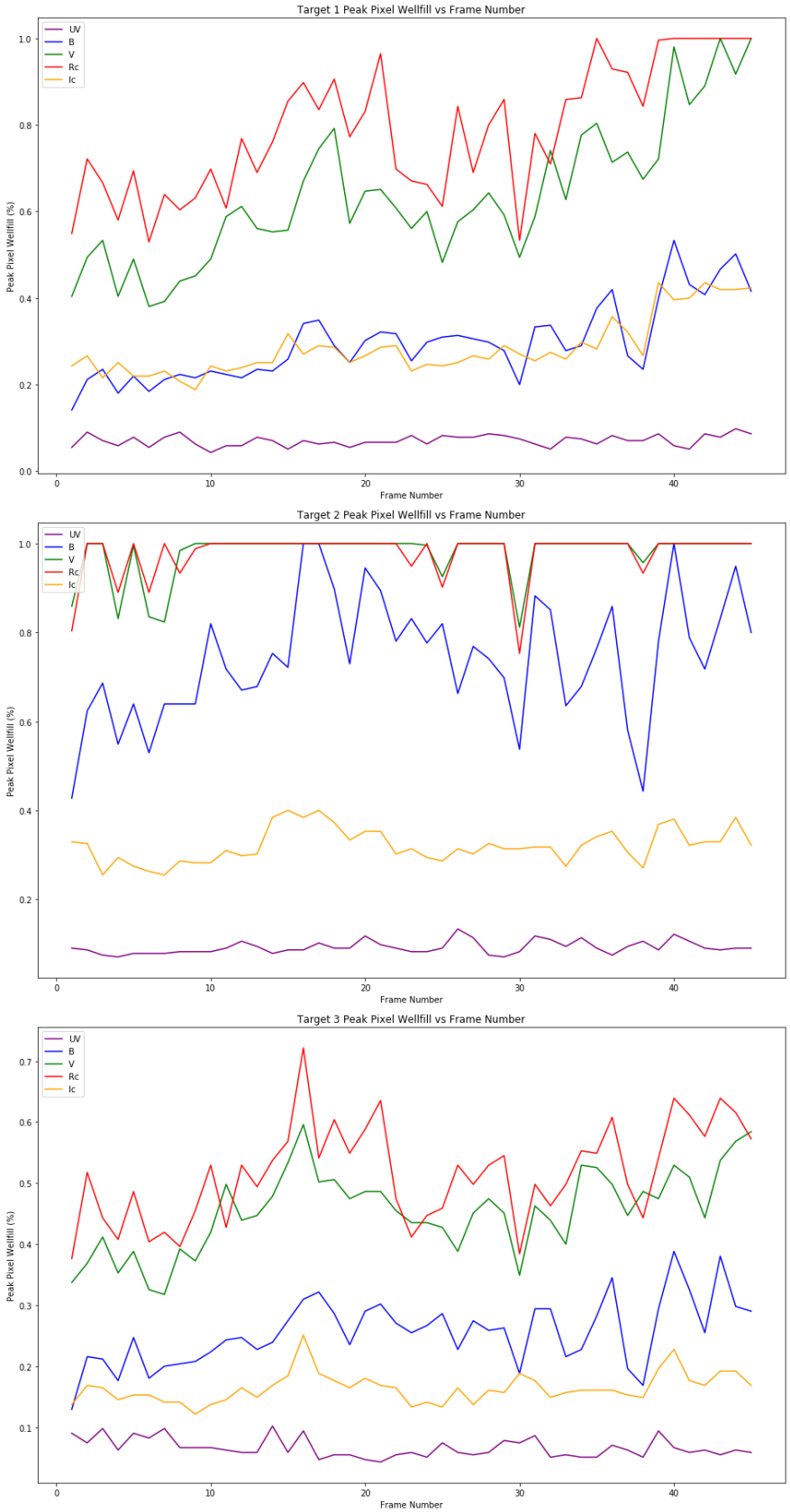

Fig. 3. Plots generated for the 3 targets observed in Fig. 2. Targets are numbered from right to left as seen in Fig. 2

8. EXTENSIBILITY

This system is designed to be as generic as possible in order to allow future extensibility. All components are built to allow hardware to be easily swapped in and out. Creating generic software inherently complicates the code. It is much simpler to create a system that controls specific hardware than to create a system that can control any hardware. For this reason, we tried to use generic interfaces such as ASCOM whenever possible. Unfortunately, not all hardware supports this general framework. In these instances, we were forced to wrap specific SDK's into our general framework. As an initial proof of concept of the extensibility of the Sohbrit system, in approximately a week's time we were able to stand up a second system using new collection hardware. This second system resides in the same dome, but incorporates a new mount and a collection of new cameras.

Sohbrit's modular design enables a fast transition to new observations. Imaging hardware can easily be swapped to capture different phenomenology. Filters can also be swapped to gain new information. Fig. 4 shows a collect on the Anik cluster, a satellite cluster commonly used for calibration, captured by Sohbrit using a hyperspectral grating. Sohbrit can also be used to collect data on other celestial objects such as nebulas or galaxies. Fig. 5 shows a processed image of M-51 collected by Sohbrit.

Fig. 4. Image of Anik cluster through hyperspectral grating

Fig. 5 Processed Image of M-51 (Whirlpool Galaxy)

9. HARDWARE LIST

We have amassed a collection of various astronomy hardware components throughout the process of creating Sohbrit. Here we list the current setup used to perform the collections relevant to the processing above.

Dome:	Astro Haven 12 ft. Clamshell dome
Mount:	Astro Physics 1600GTO German Equatorial mount
Camera:	ZWO ASI 1600MM-Cool
Auto-Focuser:	Starlight Instruments Focuser Boss II Electronic Focus Control
Filter wheel:	Starlight Xpress 7-Position
Filters:	Astrodon Johnson/Bessel: U, B, V, Rc, Ic

Fig. 6. Sohbrit facility in Albuquerque, NM

10. FUTURE WORK

Sohbrit currently schedules, collects, stores, and disseminates collections for geosynchronous characterization. We intend to expand this capability to allow Highly Elliptical Orbit (HEO), Middle Earth Orbit (MEO), Low Earth Orbit (LEO) characterizations as well. While limited rate-tracking is currently implemented, we leave more advanced automated closed-loop tracking as future work. As a national laboratory we also anticipate using Sohbrit to evaluate various SSA hardware and software components for other projects.

ACKNOWLEDGEMENTS

We would like to thank Mark Bastian for the idea and motivation to start this project. This project would not have been possible without the guidance and experience of Shane Ramotowski. We would also like to thank Kyle Merry and Liam Boone for the creation and support of the processing pipeline. Finally, we would like to thank the number of interns that have worked on this project and helped it become what it has today.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

1. Fast, Wall, and Chen, "Java Native Access", GitHub repository, github.com/java-native-access/jna, 2007.
2. Fielding, Roy Thomas, *"Chapter 5: Representational State Transfer (REST)"*. *Architectural Styles and the Design of Network-based Software Architectures,* University of California, Irvine, CA, 2000.
3. Robert B. Denny, "ASCOM: Review of the Technology and Milestones," *Minor Planet Amateur/Professional Workshop*, pp.88-92, Tucson, AZ, 2001.
4. Historic Averages: Albuquerque, New Mexico, www.intellicast.com/local/history.aspx?location=USNM0004, 2017.

5. Jerry Lodriguss, “Focus Change Due to Temperature Variation”, *Catching the Light,* www.astropix.com/wp/2011/06/20/focus-change-due-to-temperature-variation/, 2011.
6. Larry Weber, Steve Brady, “Fast Auto-Focus Method and Software for CCD-based Telescopes”, *Minor Planet Amateur/Professional Workshop*, Tucson, AZ, 104-113, 2001.
7. Lang, D., Hogg, D.W., Mierle, K., Blanton, M., & Roweis, S., “Astrometry.net: Blind astrometric calibration of arbitrary astronomical images”, *The Astronomical Journal* 139, 1782—1800, 2010.
8. Main Sequence Software, Sequence Generator Pro, mainsequencesoftware.com/content/Sequence%20Generator%20Pro.pdf, 2017.
9. Facebook, “GraphQL Specification”, facebook.github.io/graphql/, 2016.
10. D. P. Bertsekas, “A distributed algorithm for the assignment problem, unpublished working paper”, Laboratory for Information and Decision Sciences, Mass. Inst. of Technology, Cambridge, MA, 1979.