Autonomous Template Generation and Matching for Satellite Constellation Tracking

Conference Paper · January 2022

DOI: 10.2514/6.2022-0528

David Zuehlke
Embry-Riddle Aeronautical University

Madhur Tiwari
Florida Institute of Technology

Autonomous Template Generation and Matching for Satellite Constellation Tracking

David Zuehlke*
Space Technologies Lab, Embry-Riddle Aeronautical University, Daytona Beach, Florida, 32114

Madhur Tiwari†
Florida Institute of Technology, Melbourne, Florida, 32901

Daniel Posada‡ and Troy Henderson§
Space Technologies Lab, Embry-Riddle Aeronautical University, Daytona Beach, Florida, 32114

This paper proposes a method of autonomous generation of template images for satellite constellation tracking. An initial template image is formed by determining the locations of several resident space objects (RSOs) in an optical image. A template image of a single RSO (i.e. point-source) is matched against the image to determine these RSO locations. The image locations of detected RSOs are then used to form an initial template image. Once an initial template is formed, subsequent frames are matched via normalized cross-correlation to find the generated template in each image. Specific RSOs are then associated across frames by their known position in the template image. In order to handle time-varying configuration of satellites the template image is periodically updated based on a threshold on correlation, or unsuccessful matching. Orbit determination via Gooding's method is then used to provide an initial orbit estimate followed by a precise orbit estimate based on a sequential Unscented Kalman Filter. Experimental results using images obtained from a ground-based telescope system are presented to showcase the algorithm's operation.

I. Introduction

SPACE is becoming more crowded with every passing day. Current estimates put at least 700,000 objects greater than 1 cm in size orbiting the Earth [1, 2]. The vast majority of these objects are uncatalogued space debris objects caused by orbital collisions, satellite breakup events, launch events, and the latest military testing on anti-satellite weaponry [3]. The prospect of mega-constellations such as SpaceX's Starlink satellites, propose to loft thousands of satellites into low-earth orbit (LEO). To date, more than a thousand Starlink satellites have already been launched and can be seen crossing the night sky like a long string of pearls. The ever-increasing number of satellites in orbit demands that groups of satellites (constellations or swarms) must cooperate within close proximity to accomplish their missions. Providing accurate orbit estimates of these satellites relative to one another is of paramount importance for preventing on-orbit collisions. Space domain awareness (SDA) tracking data has two main contributing methods, active tracking using radar, and passive tracking using optical means. Optical tracking methods have long been used to provide space situational awareness (SSA) data for performing orbit estimation, and a variety of existing methods provide exceptional results [4]. Optical tracking involves using a ground (or space based) telescope to capture an image of one or more resident space objects (RSOs) and the background star field. There are a variety of ways of detecting which objects in a given image represent stars and which represent resident space objects (RSOs). One approach involves using the gross-motion of objects in the image sequence to discriminate between background objects (stars) and RSOs [5]. Another common method involves using streak detection [6, 7]. Depending on various observation factors such as exposure length, camera sensitivity, and tracking mode, RSOs will appear either as point-sources or as streaked objects in an image. Another method of RSO detection is to use information from the inertial frame. In an inertial frame, RSOs exhibit motion, whereas stars remain stationary. Assuming availability of sufficiently dense measurements, the motion in the inertial frame can be used to determine RSO locations from background stars and was demonstrated in Refs. [8, 9].

*PhD Student, Aerospace Engineering Department
†Assistant Professor, Department of Aerospace, Physics and Space Sciences
‡PhD Student, Aerospace Engineering Department
§Associate Professor of Aerospace Engineering, Aerospace Engineering Department

Previous work by the authors demonstrated a novel approach for identifying RSOs in an image using template matching for tracking RSOs in a series of images [10]. Given that many satellites operate in groups (constellations), template matching for the constellation solves the data association problem of determining which detected RSO associates across a series of images. However, a significant drawback of the method involved was the need for a-priori knowledge to provide a template image of the constellation to be tracked. Additionally, if the constellation exhibited relative orbital motion (i.e. the constellation changed with time), then a static template becomes invalidated after the satellites change position relative to one another. Geostationary constellations change slowly over time given their orbit geometry and the presence of long term perturbations, however, LEO constellations can exhibit much more rapid relative orbital motion due to the nature of their orbits. Dynamically changing scenes requires a method of updating the template image being matched against a target image. Using the image data itself to update the template image is possible and was demonstrated for tracking a small UAS quad-copter in flight [11].

This paper seeks to extend the concept of using template matching for satellite constellation tracking by providing a method of autonomously generating template images, as well as using a periodically updating the template image to counter the effects of changing satellite configurations. The method involves matching a single object (RSO) to multiple objects in an image to find the locations of all RSOs present in a constellation. These matched locations are then used to generate a template image for matching subsequent images. Once the constellation has been matched, an initial orbit estimate can be formed using any number of multi-measurement orbit determination (OD) methods. There exist N-measurement forms of Gauss and Gooding's methods which could be used for performing the OD step [4]. However, filtering approaches often provide the best initial orbit estimate given large numbers of observations. In [12] a batch filter using the Unscented Transform was utilized to form an orbit estimate from range and angular measurements. In this research a sequential Unscented Kalman Filter (UKF) is utilized to process multiple measurements. There has been previous work done using angles-only methods of non-linear filtering for spacecraft navigation (Ref. [13]) and the concepts applied are very similar to the problem of sequential filtering for angles-only orbit determination [14]. For this research it will be assumed that an initial orbit estimate to start the filter is available from prior knowledge, or can be generated using a three-measurement implementation of Gooding's method [15, 16]. The rest of this paper is structured as follows. First relevant background theory for template matching and orbit determination is given. This is followed by sections detailing the template matching and autonomous template generation algorithms. Followed by the implementation details of the UKF used for orbit determination, and the overall algorithm description. Results of the algorithm's use on actual ground-based space imagery are presented and finally conclusions are drawn and future work is described.

II. Background and Theory

A. Template Matching

Template matching is a well known algorithm for determining whether a query (template) image is contained in a target (base) image. Methods of template matching are varied and include robust methods such as shown in Ref. [17] where the authors proposed a method of iteratively searching through multiple possible rotation transformations of a template image. Another method is shown in Ref. [18] where an intelligent search method was used to propose scale and rotation changes. Normalized cross-correlation (NCC) is a well known method of template matching and has been shown to be robust to changes in scale and rotation [19]. Essentially the method works by comparing the template image to all the pixels of the target image to compute a matrix of coefficients representing the degree of correlation [19, 20]. The correlation map can be thought of as a 3D surface where the highest peak corresponds to the location of the template image in the target image. When the correlation coefficients are represented as a 3D surface, the peaks represent areas of close correlation. Assuming a successful match, the maximum peak gives the location of the region matching with the template image. Equation (1) is used to find the matrix of coefficients and is the heart of the template matching algorithm. Note that: $\gamma(u, v)$, is the matrix of correlation coefficients, $I(x, y)$ is the input image, $T(\bar{x}, \bar{y})$ is the template image, $\bar{I}_{x,y}$ is the input image mean, and $\bar{T}$ is the template image mean intensity.

$$\gamma(u,v) = \frac{\sum_{x,y}\left[I(x,y) - \bar{I}_{x,y}\right]\left[T(x-u,y-v) - \bar{T}\right]}{\left[\sum_{x,y}\left[I(x,y) - \bar{I}_{u,v}\right]^2 \sum_{x,y}\left[T(x-u,y-v) - \bar{T}\right]^2\right]^{0.5}} \tag{1}$$

Normalizing the correlation coefficients such that: $[-1 \le \gamma(u, v) \le +1]$, helps to overcome difficulties that plague cross-correlation such as illumination, rotation, and scale differences between the target and template image [20]. The

MATLAB implementation of `normxcorr2` was used in this research for NCC. Figures 1 and 2 show an example of template matching and the associated peak in the cross correlation surface (courtesy of MATLAB [21]). The highest peak represents the location of the template image in the target image.

Fig. 1 Example of template matching from Ref. [21]

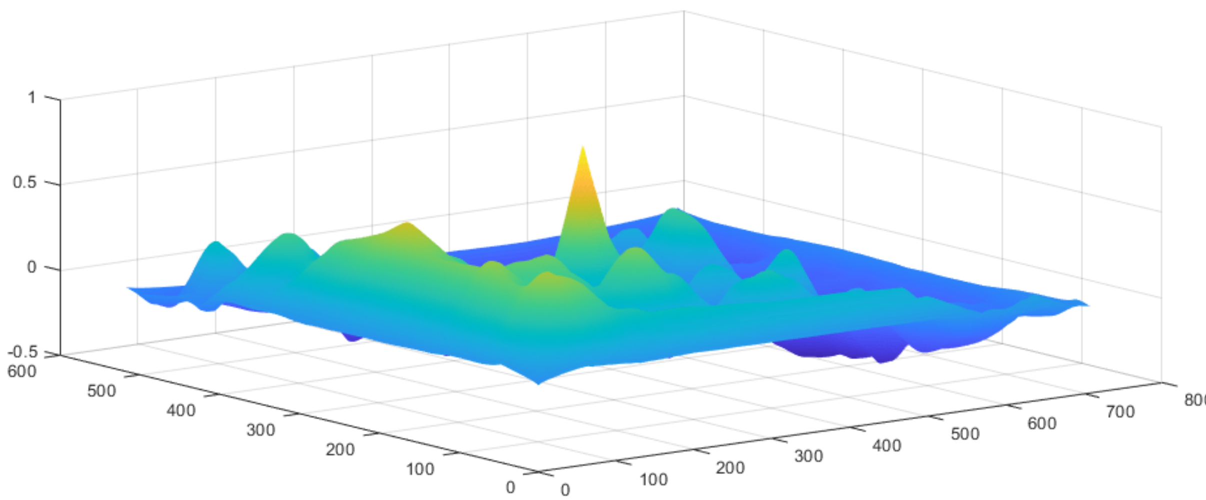

Fig. 2 Example correlation peaks from Ref. [21]

III. Autonomous Template Generation Algorithm

In this section the autonomous template generation process will be outlined. The basic idea is to use the template image of a single object (RSO) to create a correlation map in an image containing multiple RSOs. A threshold on matching value is set and all objects above the threshold are considered RSOs. From those detected locations, a template image preserving the location of all RSOs is created. Note that theoretically one could use the template of a single RSO to match for all images, however by creating a template image from a constellation of $M > 1$ RSO objects there are more feature points to correlate and matching results are improved providing more robust matching compared to using the single RSO template. Furthermore, once a template containing multiple objects is formed then the data association problem of determining which RSO associates across images is solved. Otherwise detected RSOs must be associated across frames, which in itself is a non-trivial process [10]. Figure 3 shows the outline of the autonomous template generation process. Further details are included below and summarized in algorithm 1. The first step is to load a raw

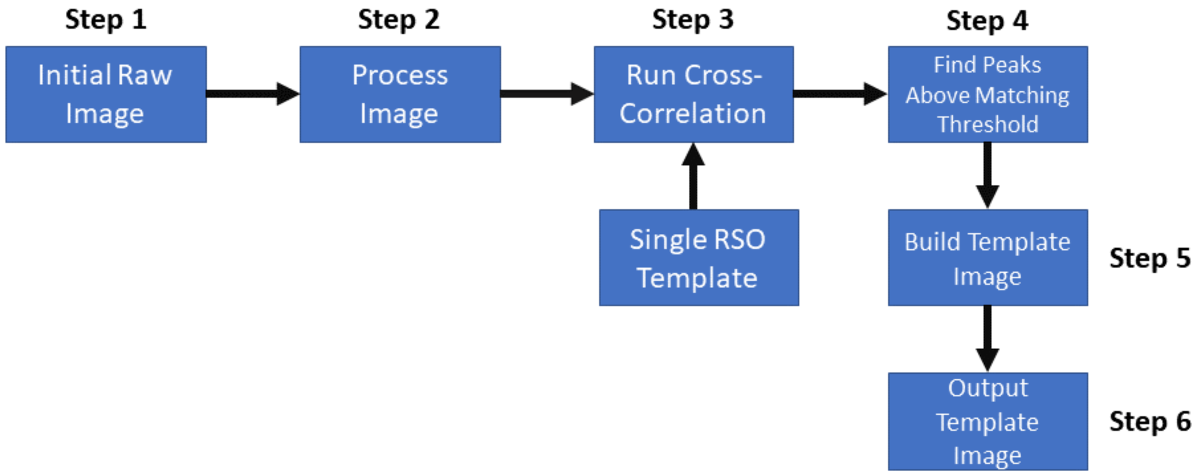

Fig. 3 Autonomous template generation process

image that contains the constellation that is desired to be tracked across a series of images. Figure 4 shows an example raw and processed image of a constellation of GEO satellites. A cropped portion is shown for clarity to highlight the visible RSOs in the raw image (left) and and the processed image (right).

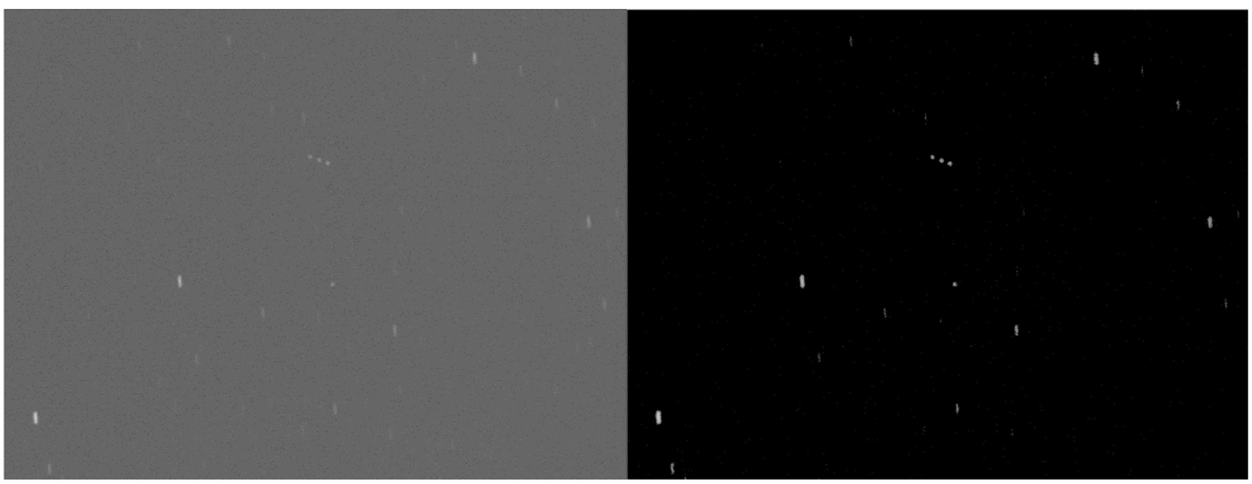

Fig. 4 Raw (left) and processed (right) cropped portions of example image containing four RSOs

Before attempting to find the RSOs in a raw image to form the template image, the image is processed to reduce noise. Gaussian smoothing, and a threshold filter are applied to smooth and then remove the background image noise. Further details in the space image processing can be found in references [9, 19]. An example processed image showing a constellation of GEO satellites is shown in the right side of Fig. 4. The processed image is then matched with an initial template, T_i of a single RSO. The initial RSO template is assumed to be a single Point Spread Function (PSF) that follows a continuous Gaussian distribution. Step 3 in the process is to run image cross-correlation on a template image of a single RSO. If the initial image $I(x, y)$ is captured such that the stars are streaking through the image and the RSOs remain stationary, then RSOs will appear as point sources in the image. Therefore cross-correlating a single-point source with the image will give strong peaks in the correlation map at all locations in the image with an RSO. The matrix of coefficients is calculated using Eq. (1) and is the heart of the cross-correlation algorithm [19]. When plotted in three dimensions, the correlation coefficients $\gamma(u, v)$ form a surface where height corresponds to the level of correlation between the base image $I(x, y)$ and the template image T. Areas of high correlation correspond to peaks in the surface and give the location of RSOs in the image. Figure 5 shows an example correlation map with peaks highlighted by red boxes and red arrows to show the locations of the single RSO template in the target image.

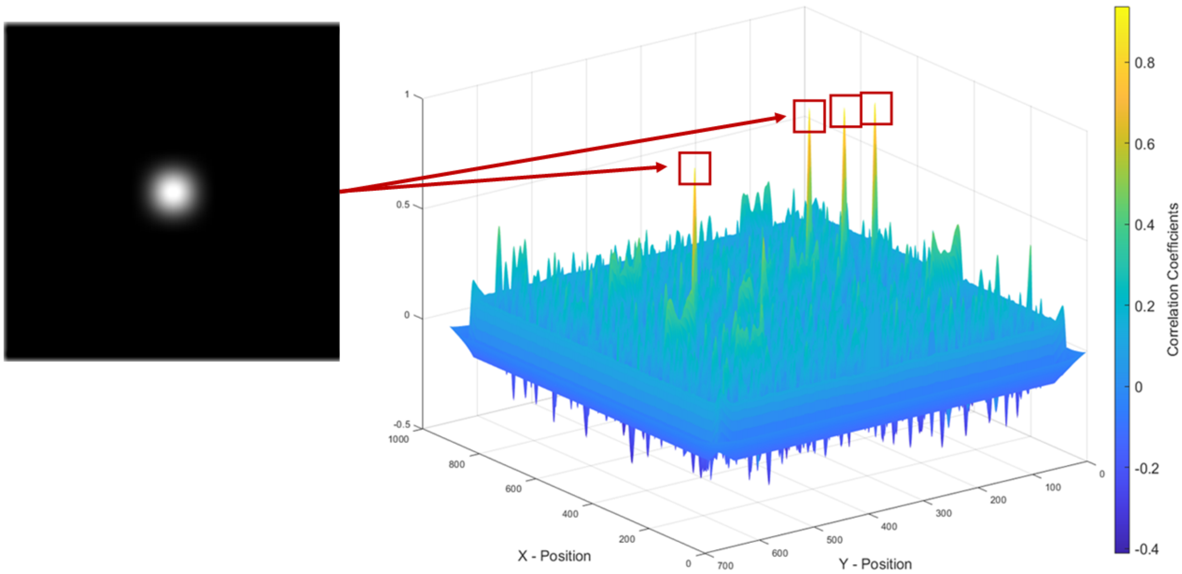

Fig. 5 RSO template matching peaks

Once the correlation coefficients have been calculated, step 4 of the algorithm is to find the set of peaks that are above a user-defined threshold. The threshold is set such that strong peaks for RSOs are kept as locations of RSOs and all other coefficients are ignored. Figure 5 shows the resulting coefficient matrix from performing cross-correlation between the processed and template images. The four highest peaks (highlighted in red) show the matched locations of the RSOs in the original image. In order to create a template image keeping the location of the RSOs, the peaks in the correlation map are found by centroiding the peak locations in $\gamma(u, v)$. Figure 6 shows the matched locations on a cropped portion of an example raw image.

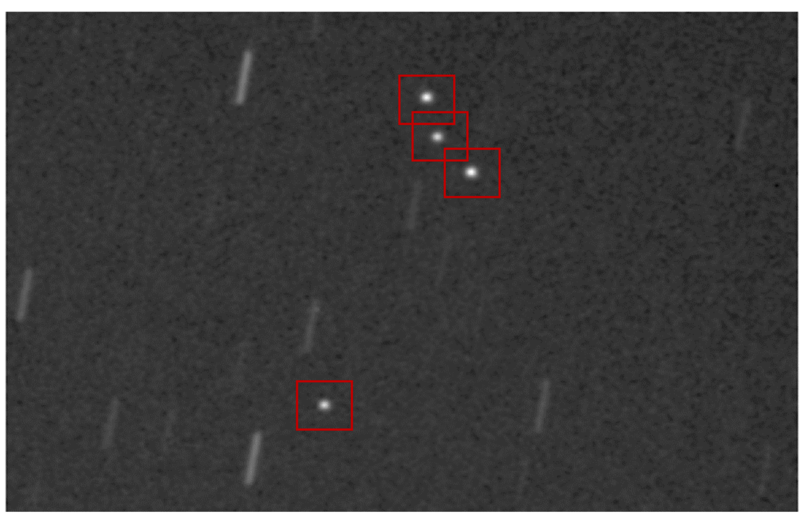

Fig. 6 Cropped raw image showing matched RSO locations highlighted in red. These locations are used to build the template image.

Now that the points of maximum correlation have been identified, the next step (step-5) is to create a template image from these matched locations. A template image is generated by first creating a matrix of all zeros with dimensions defined by a rectangular region subtended by the matched locations in the original image (with some padding pixels added around the edges to account for each RSOs PSF). The next step is to create Gaussian point sources in the template image that preserve the original detected RSO locations. An RSOs Gaussian PSF is a continuous function that is approximated by discretizing the PSF into x, y pixel coordinates. Equation 2 gives the discrete Gaussian equation where $T(x, y)$ is the template image, B is a brightness scaling factor that in practice is set to 1, σ is the Gaussian standard deviation that determines the width of the PSF, and x_c, y_c are the centroid locations of each RSO. Note that x_c, y_c are the sub-pixel coordinates corresponding to the peak locations from the correlation map $\gamma(u, v)$. Using sub-pixel coordinates for the PSF preserves the true RSO locations in the template image, and better approximates the true continuous RSO PSF in the template image.

$$T(x, y) = \frac{B}{2\pi\sigma^2} \exp\left[-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right] \tag{2}$$

For each RSO in the template image, Eq. 2 is used to add the RSOs PSF contribution to the template image. For example in Fig. 6 with four RSOs detected, equation 2 is used four times to add the contribution from each RSO to the template image. Note that in practice the Gaussian distribution in Eq. 2 falls off fairly quickly, and a maximum radius can be used to define how a cutoff distance from the RSO centroid. Algorithm 1 summarizes the steps used to autonomously generate a template image given an input image, and a base template containing a single RSO.

Algorithm 1 Generate Template Image

Require: Input base image $I(x, y)$ initial template of single RSO T_i, and matching threshold ϵ_m
  Process base image to improve detection results (Gaussian smoothing, threshold filtering)
  $I_p \leftarrow \texttt{process image}(I(x, y))$
  Run normalized cross correlation on processed image with initial template using Eq. 1
  $\gamma(u, v) \leftarrow \texttt{NCC}(I_p, T_i)$
  Find peaks above the matching threshold and set these peaks as the locations of M RSOs in the template image
  $\mathbf{X}_{RSO} \leftarrow \gamma(u, v) > \epsilon_m$
  Create template image using dimensions that encompass all elements of $\mathbf{X}_{RSO}$
  Loop over the number of RSOs (M) and add the PSF contribution from each to the template
  for $j = 1$ to M **do**
    Add RSO i to template image using Eq. 2
  end for
  return Output template image T

An example template image generated autonomously with algorithm 1 is shown in Fig. 7. Note that the template image was rotated 90 degrees purely for display purposes. Now that a template image has been created, the template matching process via NCC can be applied to all subsequent images.

A. Template Matching and RSO Association

Once a template image T has been created using algorithm 1, the template image can then be matched against target image $I(x, y)$ and detected RSOs can be associated across image frames by their respective positions in the template image. Algorithm 2 shows the process of finding a matching template location $\mathbf{X}_{matched}$ in a target image $I(x, y)$ using normalized cross-correlation (NCC). Finding the center of the template is only the first step in determining RSO locations and associating them across images. Once the template has successfully located in target image $I(x, y)$ the next step is to pull out a sub image I_s centered at the located template location. The sub image is then operated on to associate RSOs from the template image to RSOs in the actual image. Processing the smaller sub-image area corresponding to the detected template location helps to reduce the overall computational burden of the algorithm. Note that from the generated template the number of RSOs expected (M) is a known quantity. RSO locations are compared in the template image and matched sub-image area via nearest neighbor for centroids detected. Centroids are calculated using an intensity weighted centroiding method. The Euclidean distance between centroids is used to determine if a point is a nearest neighbor. For two centroid locations $\mathbf{X}_i$, $\mathbf{X}_j$ given by x, y coordinates the distance is given by the Euclidean norm:

$$d = \|\mathbf{X}_i - \mathbf{X}_j\| \tag{3}$$

Fig. 7 Autonomously generated template image example. (Rotated from actual orientation for convenience)

Algorithm 2 Template Match Image

Require: Input image $I(x, y)$, template image T, matching threshold ϵ_m
  Process base image $I(x, y)$ to improve detection results
  $I_p \leftarrow \texttt{process image}(I(x, y))$
  Run normalized cross correlation on processed image with current template image using Eq. 1
  $\gamma(u, v) \leftarrow \texttt{NCC}(I_p, T)$
  Find template location from maximum peak
  $\mathbf{X}_t \leftarrow \texttt{max}(\gamma(u, v) > \epsilon_m)$
  return $\mathbf{X}_t = [x_t, y_t]^T$ where x_m, y_m denotes the coordinates of the template in image I
Ensure: Located template at $\mathbf{X}_t$ lies within image $I(x, y)$

Two points are nearest neighbors if $d < \epsilon$ for a given threshold in pixel distance ϵ. Given a set of centroids consisting of x, y pairs in pixel space, finding nearest neighbors reduces to finding the minimum distance pairs between two sets of points. Centroiding the target sub-image centered on the template location gives one set of centroids $\mathbf{X}_s$ where the s subscript denotes the sub-image. Centroids from the template image are denoted $\mathbf{X}_T$ and can be calculated one time when the template image is created. In order to provide consistent labeling of RSOs in the template image, RSO centroids are labeled with an index number starting from the top left of the image and proceeding to the bottom right of the image in pixel space. Once both target and template image centroids have been found, nearest neighbors are found with the minimum distance pairs below a threshold level (the threshold prevents a non-detection of an RSO from associating incorrectly to a star or noise point that was the closest point but clearly not the RSO). To associate RSOs loop through number of RSOs in the template image (M), and find the nearest neighbor matching to each RSO. If the j^{th} centroid $\mathbf{X}_{T_j} = [x_{s_j}\ ,\ y_{s_j}]$ is the nearest neighbor given by:

$$NN = min(d(\mathbf{X}_s - \mathbf{X}_{T_j}) \tag{4}$$

Then the j^{th} matched RSO location is associated successfully, i.e. $\mathbf{X}_{matched_j} = \mathbf{X}_s$. If no nearest neighbor is found, then the j^{th} RSO was not matched in this image, and the loop continues for the next RSO. The process outlined above for associating RSOs is summarized in algorithm 3.

B. Template Update Process

Over time the relative orbital motion (or maneuvering) of constellation agents can cause the original template image to become outdated. In order to overcome this difficulty, a process of updating the original template image based on the current detected location of RSOs is implemented. Reference [11] proposed a method of updating template images in dynamic video scenes. The process is similar to the case of a changing satellite constellation. In the above mentioned paper, the template image of a small UAS quadcopter was updated to track the object in flight. Large changes

Algorithm 3 Associate RSOs

Require: Template matched location $\mathbf{X}_t$, processed image $I_p(x, y)$, template image T, and pixel matching threshold for RSOs ϵ_d

Pull out sub-image $I_s(x, y)$ from $I(x, y)$ centered at $\mathbf{X}_{matched}$ with dimensions $\mathbb{R}^{p\times q}$ the dimensions of T
Detect centroids in $I_s(x, y)$ and T
$\mathbf{X}_s \leftarrow \texttt{centroids}(I_s(x, y))$
$\mathbf{X_T} \leftarrow \texttt{centroids}(T)$
for $j = 1$ to M **do**
  Find nearest neighbor to RSO locations and associate that centroid as RSO_j
  $\mathbf{X}_{T_j} \leftarrow x, y$ centroid of RSO_j
  if $\texttt{min}(d(\mathbf{X}_{ks} - \mathbf{X}_{T_j}) < \epsilon_d)$ **then**
    Nearest neighbor is associated successfully
    $[x_{s_j}, y_{s_j}] = \mathbf{X}_{T_j}$
    $\mathbf{X}_{matched_j} = [x_{s_j}, y_{s_j}]$
  else
    $\mathbf{X}_{T_j} \leftarrow$ Not matched for current image
  end if
end for
return $\mathbf{X}_{matched_j}$ All matched centroid locations

in illumination, orientation, and size prompted the need for a template update. With satellites constellations, the changes will be presumably be on a much slower scale, but the same concept applies. There are two criteria for determining if a template update is necessary. The first criteria compares the peak correlation value $\gamma_{max} = \texttt{max}(\gamma(u, v))$ to a defined matching threshold ϵ_m. Note that the matching threshold depends on several factors including image noise. For this research a value of $\epsilon_m = 0.35$ provided a good balance between update frequency and matching accuracy and was determined empirically. If $\gamma_{max} < \epsilon_m$, then we consider the match to be of too poor quality to attempt association of RSOs and the template image is updated from the previously successful matching image. Once the template has been updated, a second attempt is made to match the current image $I_k(x, y)$ to the new template. The second criteria for updating the template image is if association of RSOs fails. Failed association could occur for a number of reasons, including image noise, poor image quality, or failure to find the inertial coordinates of the RSOs from background image stars. If association does fail the template image is updated from the previous successful matching image and a second attempt is made to match the image and associate RSOs. Once criteria have been met to update the template image, the same process (algorithm 1) used to generate the initial template T is used to generate a new template $T_{updated}$ from the last image that successfully matched. Algorithm 4 outlines the process of determining if a template update is necessary, and then the process of updating the template image. At this point all processes necessary to the optical

Algorithm 4 Template Update Process

Require: Current matched correlation value γ_{max}, matching threshold ϵ_m

if $\gamma_{max} < \epsilon_m$ **then**
  Update template using algorithm 1
  $T_{updated} \leftarrow \texttt{generate template}(I_{t_{k-1}}, T_i)$
else if $\gamma_{max} > \epsilon_m$ but no RSOs associated **then**
  Update template using algorithm 1
  $T_{updated} \leftarrow \texttt{generate template}(I_{t_{k-1}}, T_i)$
end if

tracking via template matching and association of RSOs are complete. The end goal of the tracking is to estimate the orbit of detected RSOs and is detailed in the next section.

IV. Orbit Estimation

Obtaining accurate estimates of a satellite's orbit is of paramount importance. Classical orbit determination techniques such as the methods of Gauss, Laplace, and more recently Gooding's method provide the foundation for modern orbit estimation techniques [4, 22]. The autonomous template matching method of this research lends itself well to a sequential orbit estimation method since measurements are not restricted to be taken at any specific time and there are no limits on the total number of observations. Batch methods of orbit estimation could also be used given the whole set of observations to process at once. Reference [23] proposed a batch unscented estimation method for optical orbit determination. Batch methods of orbit determination are the subject of future work. In this research an initial orbit estimate is formed via Gooding's method and then a sequential Unscented Kalman Filter (UKF) is used to refine the orbit estimate from the set of all angular measurements [16, 22, 24]. The UKF has the advantage of accurately representing the true non-linear dynamics of a system and avoiding the computationally intensive computation of partial derivatives for the system needed for both batch least-squares and Extended Kalman Filter (EKF) estimators. These advantages have been shown for a number of applications and the same is true for orbit determination [22]. The UKF formulation allows the user to choose the desired fidelity for the propagation model. For this research two-body motion is deemed sufficiently accurate as the propagation force model due to measurements being spread at most a few hours apart. Depending on the accuracy required, the propagation technique can easily be extended to include orbital perturbations such as J2, atmospheric drag, and solar radiation pressure, or even to a full ephemeris model. The UKF requires an initial state estimate to begin the estimation process. It is assumed that an initial state estimate is available from either previous knowledge or as the result of an initial orbit determination (IOD) technique such as Gooding's method. In this research an initial state estimate is be taken from available Two-Line Element (TLE) data in order to test the UKF algorithm under the best possible circumstances. Future work will include seeding the UKF with an initial orbit solution from Gooding's method.

A. Initial Orbit Determination

The end goal of obtaining optical images of RSOs is to find an orbit estimate for the observed line-of-sight (LOS) vectors of an observed RSO. In order to accomplish this, the camera frame centroid (i.e. x, y location) of an RSO must be transformed into an inertial LOS vector from the observer's location to the RSO. When capturing images of RSOs, the background star field is used to obtain the inertial LOS vector through a process known as plate-solving. Plate solving uses the observed positions of stars and compares them to a known star catalog to obtain the inertial pointing direction of the camera. Astrometry.net is a program that provides robust plate-solving capabilities for space imagery that will calculate the inertial (Right Ascension (RA) and Declination (DEC)) angles of an image given the position of the stars in the image and basic camera parameters [25]. Once RA, DEC coordinates are available for all detected RSOs then it is a simple matter to convert the angles into LOS vectors using Eq. 5 [22].

$$\widehat{\mathbf{L}}_k = \begin{bmatrix} cos(\delta_{t_k})cos(\alpha_{t_k}) \\ cos(\delta_{t_k})sin(\alpha_{t_k}) \\ sin(\delta_{t_k}) \end{bmatrix} \tag{5}$$

Where $\widehat{\mathbf{L}}_k$ is the k^{th} LOS vector at time t_k and α_{t_k} and δ_{t_k} are the corresponding RA and DEC angular measurements.

Given the time of observation, LOS vectors, and observer position, the problem is now to fit the observations to an orbit. Classically the problem is solved given observations at three times because the measurement consists of two angles at each time which corresponds to the minimum number of observations to find an orbiting objects six orbital states [22]. Gooding's method of IOD will only be summarized here, for further details of the implementation of Gooding's method see references [16, 26]. Gooding assumes that measurements are available at three times, and also requires a guess for the range to the satellite at each time. Then Gooding assumes that the range estimates to the first and last measurements are perfect placing all the "error" on the middle measurement. The orbit is then estimated from the first and last measurements using a Lambert solver and a Newton Raphson method that corrects all three ranges until convergence criteria are met. The result is the orbital position and velocity at the time of the first observation. This initial orbit estimate can be used as the starting point for the UKF orbit estimation method presented in the following section. Note that Gooding's method provides no measure of the uncertainty of the IOD estimate.

B. Unscented Kalman Filter for Orbit Estimation

Given a time series of angular observations (the output of the template matching algorithm) a UKF is then used to estimate the orbit of each member of the detected constellation at each time step. The unscented Klman filter is a sequential filtering method that begins with an initial state and covariance estimate. First, define the desired state vector of the satellite as the satellite's position and velocity vectors $\mathbf{X} \in \mathbb{R}^6$.

$$\mathbf{X} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{6}$$

The UKF works by taking a set of statistically important points called Sigma Points (denoted by χ_i) that preserve the mean and covariance of an initial distribution of points. In order to preserve the mean and covariance, a minimum of $2N$ sigma points are required where N is the dimension of the state vector ($N = 6$ for the position and velocity states of a satellite). Assumed errors in the states are used to generate the initial covariance and are set as $\epsilon_r = 10km$ and $\epsilon_{\dot{r}} = 1m/s$ as the position and velocity a priori error estimates. In order to propagate the sigma points forward in time, the nonlinear dynamics of the system are defined as the derivative of the state vector. Where μ is earth's gravitational parameter and $r = \|\mathbf{r}\|$ is the magnitude of the position vector given by $\mathbf{r} = [x\ y\ z]^T$.

$$\mathbf{f}(\mathbf{X}) = \dot{\mathbf{X}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & -\frac{\mu x}{r^3} & -\frac{\mu y}{r^3} & -\frac{\mu z}{r^3} \end{bmatrix}^T \tag{7}$$

The UKF requires a set of $2N$ statistically important points to approximate the first two moments of given distribution, where $N = 6$ is the number of states for a satellite [24]. The selection of the so called sigma points can be accomplished by choosing a set of $2N$ points such that the mean and covariance of the original states is maintained. A set of weighting parameters on the sigma points with a tuning parameter α are set such that the desired distribution is achieved. A value of $\alpha = 0.95$ was chosen through for this research. Begin by setting $L = 2N$ to be the number of sigma points. Then define auxiliary constants $\kappa = 3 - L$, and $\lambda = \alpha^2 (L + \kappa) - L)$ as scaling factors to define the spread of the distribution around the mean. Where once again α is a free tuning parameter that typically lies in the range $0 < \alpha < 2$. Then the ith sigma point denoted by χ_i is calculated as:

$$\chi^{(i)} = \mathbf{X}_k \pm \sqrt{(L+\lambda)P_k)}_i^T \tag{8}$$

Where $\mathbf{X}_k$ is the mean at time t_k, P_k is the covariance at t_k and $\sqrt{(L+\lambda)P_0)}_i^T$ denotes the ith column of the matrix square root (in practice the Cholesky decomposition). The dimensions of P_k are $\mathbb{R}^{6\times 6}$ and to form $2N$ points each column is taken on either side of the mean of the state. Once a set of sigma points is established at the initial time t_0 the sequential part of the filter can begin. The initial state X_0 is used to initialize the covariance at time t_0 as P_0. Once the initial covariance and state are set, the sigma points are propagated forward in time to the next measurement time through the nonlinear function $\mathbf{f}(\mathbf{X})$, defined by the state propagation model given in Eq. 7.

$$\chi^i_{t_k} = \mathbf{f}(\chi_{t_{k-1}})^i \tag{9}$$

Once the sigma points are propagated to the next measurement time, the mean of the propagated states ($\widehat{\chi}$) is found by as by averaging the propagated sigma points as:

$$\widehat{\chi}^-_{t_k} = \frac{1}{2N}\sum_{i=1}^{2N} \chi^{(i)}_{t_k} \tag{10}$$

Where the minus superscript denotes that this mean is calculated after the propagation step, but before the measurement update and the $(i)^{th}$ superscript denotes the i^{th} sigma point. From the propagated states and new mean, the covariance before the measurement also needs to be calculated and is found through equation 11.

$$P^-_k = \frac{1}{2N}\sum_{i=1}^{2N} \left[\chi^{(i)}_{t_k} - \widehat{\chi}^-\right]\left[\chi^{(i)}_k - \widehat{\chi}^-\right]^T + Q \tag{11}$$

With the mean and covariance at the current measurement time t_k, new sigma points incorporating the updated covariance are found to find the predicted measurement. Where the measurement model is given given by the calculation of topocentric right ascension (RA) and declination (DEC) angles. First the relative vector between the site and the predicted position of the RSO is found as $\boldsymbol{\rho}_k = \chi_k^{(i)} - \mathbf{r}_{site_k}$. Next calculate the LOS vector as $\widehat{\mathbf{L}}_k = \dfrac{\boldsymbol{\rho}_k}{\|\boldsymbol{\rho}_k\|}$. From the LOS vector $\widehat{\mathbf{L}}_k$ the topocentric RA and DEC denoted α_k and δ_k are calculated as:

$$\alpha_k = \mathtt{atan2}(\widehat{\mathbf{L}}_k(2), \widehat{\mathbf{L}}_k(1)) \tag{12}$$

$$\delta_k = \mathtt{asin}(\widehat{\mathbf{L}}_k(3)) \tag{13}$$

$$\mathbf{h}(\mathbf{X}_k, t_k) = [\alpha_k \quad \delta_k]^T \tag{14}$$

Where `atan2()` is the four quadrant inverse tangent function, `asin()` is the inverse sine function and where $\widehat{\mathbf{L}}_k(i) \quad i = 1, 2, 3$ give the x, y, z components of the estimated LOS vector. And finally where $\mathbf{h}$ represents the nonlinear measurement function. Now that the measurement function has been defined, the next measurement update step of the UKF can be accomplished. First measurements $\hat{\mathbf{y}}_k^{(i)}$ are calculated for all sigma points through the measurement function $\mathbf{h}(\chi_k^{-(i)}, t_k)$, followed by finding the mean measurement $\bar{\mathbf{y}}_k$.

$$\chi_k^{-(i)} = \widehat{\chi}_k^- \pm \sqrt{\left(L + \lambda)P_k^-\right)}_i^T \tag{15}$$

$$\hat{\mathbf{y}}_k^{(i)} = \mathbf{h}(\chi_k^{-(i)}, t_k) \tag{16}$$

$$\bar{\mathbf{y}}_k = \frac{1}{2N} \sum_{i=1}^{2N} \left[\hat{\mathbf{y}}_k^{(i)}\right] \tag{17}$$

$$\tag{18}$$

The measurement covariance P_y is then calculated given the measurement noise R_k (Eq. 19). Next the measurement and state cross covariance matrix P_{xy} is calculated using the sigma points and their mean, and the estimated measurements and their mean.

$$P_y = \sum_{i=1}^{2N} \left[\hat{\mathbf{y}}_k^{(i)} - \bar{\mathbf{y}}_k\right] \left[\hat{\mathbf{y}}_k^{(i)} - \bar{\mathbf{y}}_k\right]^T + R_k \tag{19}$$

$$P_{xy} = \sum_{i=1}^{2N} \left[\chi_{t_k}^{(i)} - \widehat{\chi}_k^-\right] \left[\hat{\mathbf{y}}_k^{(i)} - \bar{\mathbf{y}}_k\right]^T \tag{20}$$

$$\tag{21}$$

In order to update the state and covariance with the measurement information, the Kalman gain must first be calculated.

$$K = P_{xy} P_y^{-1} \tag{22}$$

After the measurement step, the state is updated using the Kalman gain, and the difference in the true measurement $\tilde{\mathbf{y}}_k$, and the mean estimated measurement $\bar{\mathbf{y}}_k$. The state covariance is also updated with the measurement data using the Kalman gain as shown in Eq. 24.

$$\widehat{\chi}_k^+ = \widehat{\chi}_k^- + K\left[\tilde{\mathbf{y}}_k - \bar{\mathbf{y}}_k\right] \tag{23}$$

$$P_k^+ = P_k^- + K P_y K^T \tag{24}$$

Where the "+" superscript denotes the state and covariance after the measurement update has been applied. The entire measurement and update step encompasses equations 15 - 24. After the update step the new mean and covariance ($\widehat{\chi}_k$ and P_k^+) are used to calculate sigma points and are then propagated to the next measurement time and the process continues as before. The process is summarized in algorithm 5.

The result of running the UKF algorithm on a set of RA, DEC measurements of an RSO is the state and covariance of the RSO at each measurement time. Examples of the UKF running on simulated and actual data are provided in the results section.

Algorithm 5 Unscented Kalman Filter for Orbit Estimation

Require: Initial state $\mathbf{X}_0$, covariance P_0, and set of measurements $\tilde{\mathbf{y}}_k$

for $k = 1$ to the number of measurements (N_{images}) **do**

$\chi^{(i)}_{t_{k-1}} \leftarrow \widehat{\chi}_{t_{k-1}}$ Calculate $2N$ sigma points at time t_{k-1} using Eq. 8

$\chi^{i}_{t_k} \leftarrow$ Propagate sigma points $\chi^{(i)}_{t_{k-1}}$ to measurement time t_k using Eq. 9

$\widehat{\chi}^{-}_{t_k}$, $P^{-}_{k} \leftarrow$ Find new mean and covariance from propagated states using equations 10 - 11

$\hat{\mathbf{y}}^{(i)} \leftarrow$ Generate measurements for each sigma point using equations 14 - 17

$K \leftarrow P_{xy}P^{-1}_{y}$ Find Kalman gain using equations 19 - 22

$\widehat{\chi}^{+}_{t_k}$, $P^{+}_{k} \leftarrow$ Update state and covariance using the Kalman gain and actual measurements using equations 23 - 24.

Set new mean and covariance to calculate sigma points for next iteration.

$\widehat{\chi}_{t_{k-1}} \leftarrow \widehat{\chi}^{+}_{t_k}$ and $P_{k-1} \leftarrow P^{+}_{k}$

return Mean orbital states and covariance from each iteration

$\mathbf{X}_{out} \leftarrow \widehat{\chi}^{+}$

$P_{out} \leftarrow P^{+}_{k}$

Continue to next iteration $k = k + 1$

end for

V. Autonomous RSO Constellation Orbit Determination Algorithm

Now that all the individual pieces have been discussed the full algorithm from raw image input to a final orbit estimate is presented. The end goal of the method is to provide an orbit estimate for all RSOs in a given image set. The algorithm is general in that the user need only input an image set and appropriate threshold's and the rest will be performed autonomously. Note that images are assumed to be in the Flexible Image Transport (.FIT) format with an appropriate data header that contains image metadata such as image sensor pixel size, exposure times, and image capture times. Also, note that the current implementation of the algorithm will process all images through the template matching portion, and then proceed to the UKF orbit estimation process. These steps can be easily combined into a single loop such that the UKF estimate is updated as each image is processed. The overall process to generate an orbit estimate given an input set of images is outlined in algorithm 6.

VI. Results

The full autonomous template matching algorithm was tested on a set of ground based imagery of the geostationary object AMAZONAS-2 (NORAD ID 35942) and it's surrounding satellites. A total of 1095 images were captured over a 2 hour period, where 745 images were processed through the template matching algorithm (the remaining images were thrown out due to clouds blocking the telescope field of view).

A. Experimental Setup

The imaging system consisted of a Celestron RASA 11 inch telescope with an ASI 1600m monochrome cooled camera mounted on a Celestron CGE Pro mount. The Celestron RASA 11 is an optically fast ($f/2.2$) and short focal length scope (focal length is $620mm$) [27]. These characteristics give the telescope a wide field of view (FOV) compared to other 11 inch telescopes and with the paired camera covers an area of sky approximately 1.5×1.5 degrees. The fast and wide optics are ideal for capturing the short exposure images needed to capture RSOs in various orbital domains (LEO, HEO, GEO). Additionally, imaging such a large area of sky allows the capture of multiple RSOs in each image frame. All images were processed on a desktop computer with an AMD Ryzen 5 2600 CPU with 32 Gb of RAM and a 1Tb SSD. The average solution time per image was approximately 10 seconds using non-optimized MATLAB code. Future work will investigate optimizing MATLAB code to utilize GPU arrays and parallel processing to improve run time.

B. Template Matching Results

Algorithm 6 was utilized to process all images. An initial template image was autonomously generated from the first image in the data set. The correlation map $\gamma(u, v)$ used to generate the initial template image is shown in Fig. 8. Figure 8 (a) shows all of the correlation map values plotted as a 1D array, while Fig. 8 (b) shows the 3D peaks of the

Algorithm 6 Autonomous Template Matching with UKF Orbit Estimation

Require: Input image set with N images, template image of single RSO T_i, matching threshold ϵ_m, association distance threshold ϵ_d

Generate template image from first image using algorithm 1
$T \leftarrow$ `generate template`$(I_1(x,y), T_i)$
for $k = 1$ to N **do**
  Solve template match of image I_k with NCC using algorithm 2
  $\mathbf{X}_t, \gamma_{max} \leftarrow$ `template match`(I_k, T, ϵ_m)
  if $\gamma_{max} < \epsilon_m$ **then**
    Update template image using algorithm 4
    $T \leftarrow$ `update template`(I_{k-1}, T_i)
  end if
  Associate RSOs from template image T to matched location $\mathbf{X}_t$ using algorithm 3
  $\mathbf{X}_{matched} \leftarrow$ `associate RSOs`$(\mathbf{X}_t, I_k, T, \epsilon_d)$
  if Association Failed **then**
    Update template image using algorithm 4
    $T \leftarrow$ `update template`(I_{k-1}, T_i)
  end if
end for
return Output the associated measurement sets $\tilde{\mathbf{y}}_j$ for each detected RSO
Run UKF on measurements of each RSO
for $j = 1$ to M **do**
  Generate initial orbit estimate using Gooding's method
  $\mathbf{X}_0 \leftarrow$ `gooding`$(\ldots)$
  Run UKF using algorithm 5 for RSO$_j$
  $\mathbf{X}_{out_j}, P_{out_j} \leftarrow$ `UKF`$(\mathbf{X}_{0_j}, P_{0_j}, \tilde{\mathbf{y}}_j)$
end for
return Orbital state information for all RSOs

correlation map. The peaks of the six RSOs contained in the image are highlighted by red circles in both the 1D and 3D representations. The x, y coordinates of these peaks (shown in Fig. 8 (b)) are the points used to build the initial template image. Note that the two views are shown to illustrate the advantage of building a template using multiple points rather than using only a template with a single RSO PSF (further details given below, see Fig. 11).

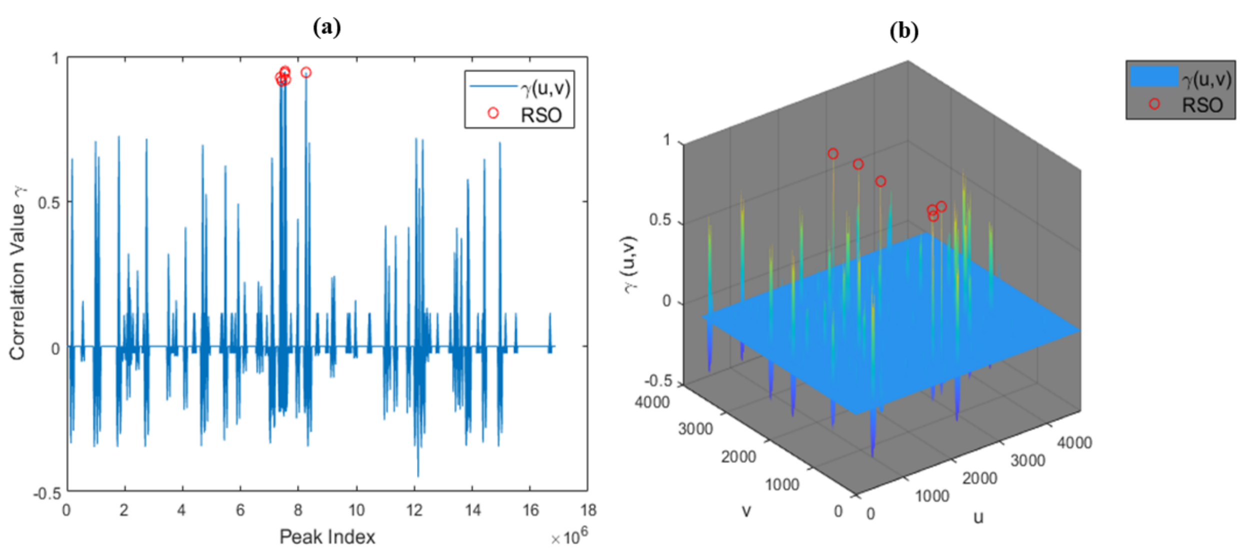

Fig. 8 (a) Correlation values in a 1D array with RSOs in red. (b) Correlation 3D map with RSOs in red

The template image that was autonomously generated is shown in Fig. 9. Note that for display purposes only the template has been rotated 90° from it's actual orientation. This template image was then matched to the proceeding in the data set until criteria were met to update the template image. One example of successful matching is shown in Fig.

Fig. 9 Rotated template image

10. The associated positions of RSOs in the image are marked with white arrows and text labels. Note that the template RSO locations shown in Fig. 9 correspond directly to the associated positions in the matched image. The corresponding correlation map $\gamma(u, v)$ in 1D and 3D formats are shown in Fig. 11. When the template containing multiple RSOs is used to match against the image, note the clear single peak corresponding to the template location unlike the multiple peaks for the case of generating the template image. This dominant peak increases the accuracy of finding the template location in the target image and is the chief advantage of using a template image containing multiple points.

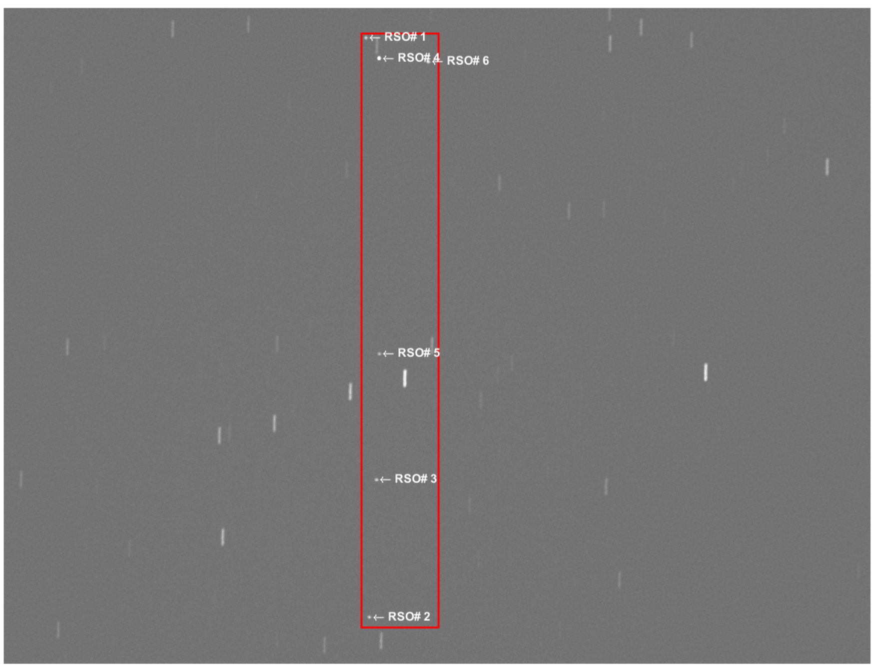

Fig. 10 Successful template match

In order to show the process of when template updates occur, the peak correlation value matched for each image is shown plotted against time in Fig. 12. The red circles denote points in time where criteria where met to update the template image. Once the template was updated, the image that failed to match before was then matched with the updated template. Note that by recursively attempting to resolve an image after the template was updated, a successful match was achieved for nearly all images. The exception were a handful of images where clouds blocked the field of view. At the points where template updates occur, there is a corresponding increase in the peak correlation value. The decrease in the peak correlation value as time increases results from motion of the satellites with respect to one another. Figure 13 shows the top three satellites in the template image at four different times to illustrate the change in position of satellites in the template image. In the span of two hours during data collection, there is a clear difference in the RSO locations in this portion of the template and showcases the need for periodic updating of the template image. Figure 13 also illustrates the successful autonomous update process for the tested image data set.

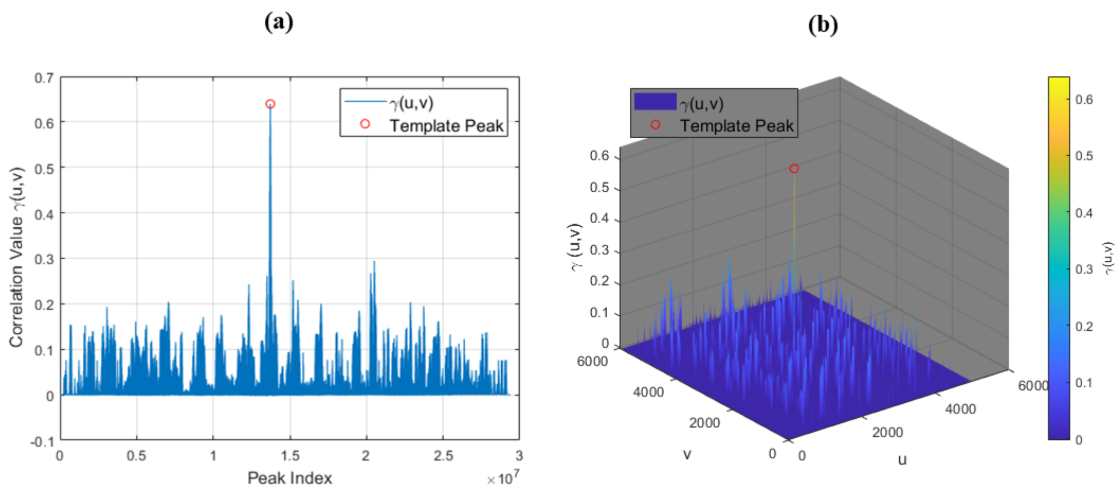

Fig. 11 Template match correlation maps

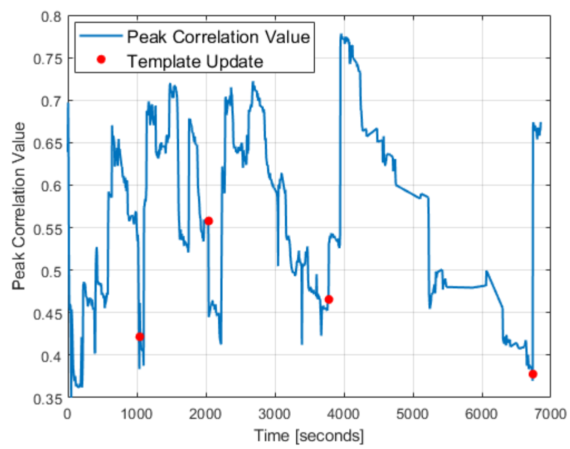

Fig. 12 Peak correlation value with template updates

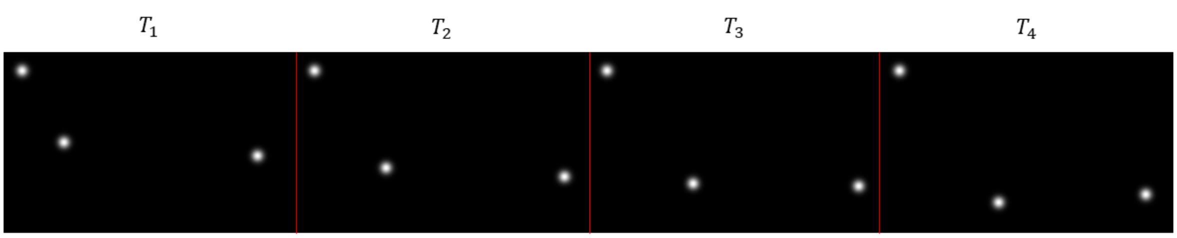

Fig. 13 Template images through time

C. Orbit Estimation Results

In order to run the orbit determination algorithm, the camera frame (x, y) coordinates of all RSOs must be converted to inertial angular observations. After a successful template match the image x, y coordinates for the RSOs are converted to Right Ascension (RA) and Declination (DEC) observation angles through a process known as plate solving. Plate solving uses the known positions of background stars in an image to find the inertial angular (RA, DEC) pointing of the camera. This research utilized the plate solving capabilities of a local installation of Astrometry.net [25, 26]. Converting all associated RSO points to RA, DEC angular observations gives the full set of measurements for orbit determination.

For the GEO objects observed over a two hour window, these observations form a set of quasi-linear trails or “tracklets”. Figure 14 shows the RA, DEC coordinates of the six RSOs detected across the image set. Each of the six RSOs was successfully associated across the image set and is shown in a unique color. The gaps in the RA, DEC trails are the result of only 745/1095 of the total images taken being processed through the template matching algorithm due to poor image quality caused mostly by cloud cover. However, the non-continuous data provides a more realistic observation scenario than simulated continuous measurements. The RA/DEC measurements shown and their respective time values are used for the orbit estimation step. Fig. 15 shows for RSO #1 the comparison of the actual measurements and UKF

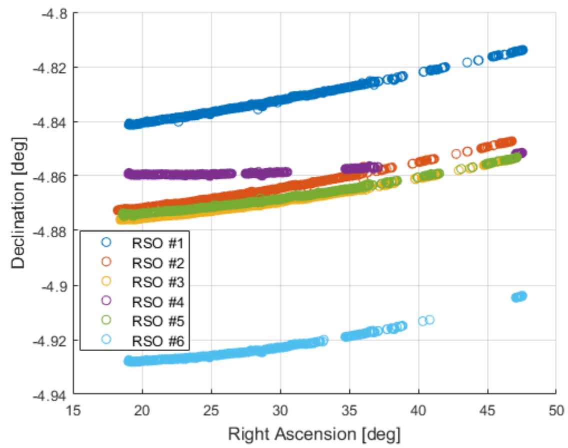

Fig. 14 RSO angular measurements

predicted measurements with their respective errors. Note that observation errors for DEC stay on average constant near 0.001° and for RA the error increases for some time and then levels out. The error in observation and predicted measurements could be caused by several factors. One possible factor is the use of only two-body propagation in the UKF, thus ignoring any perturbations affecting the actual satellite. Another possibility is that because the observation window is only 2 hours, corresponding to observing only 1/12 of the actual orbit (given a GEO satellite’s orbital period is 24 hours). Given spacing between observations the filter prediction may converge more closely to the exact solution. GEO observability is inherently low due to the fact that the orbits are nearly circular, equatorial orbits and present challenging observational geometry.

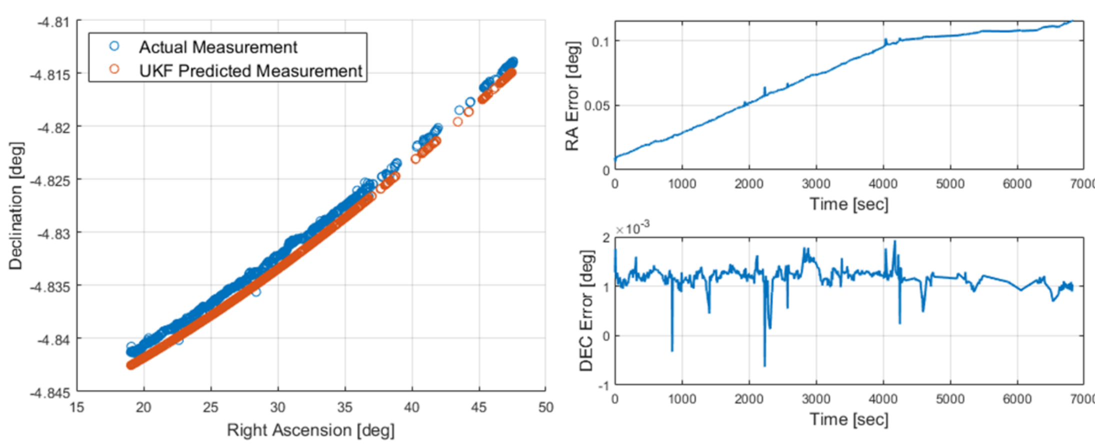

Fig. 15 UKF measurement comparison RSO#1

The UKF outlined in algorithm 5 was used to process the angular measurement data for each RSO detected in all images. The position and velocity of each RSO were predicted for each RSO at each time step and are shown in Fig. 16.

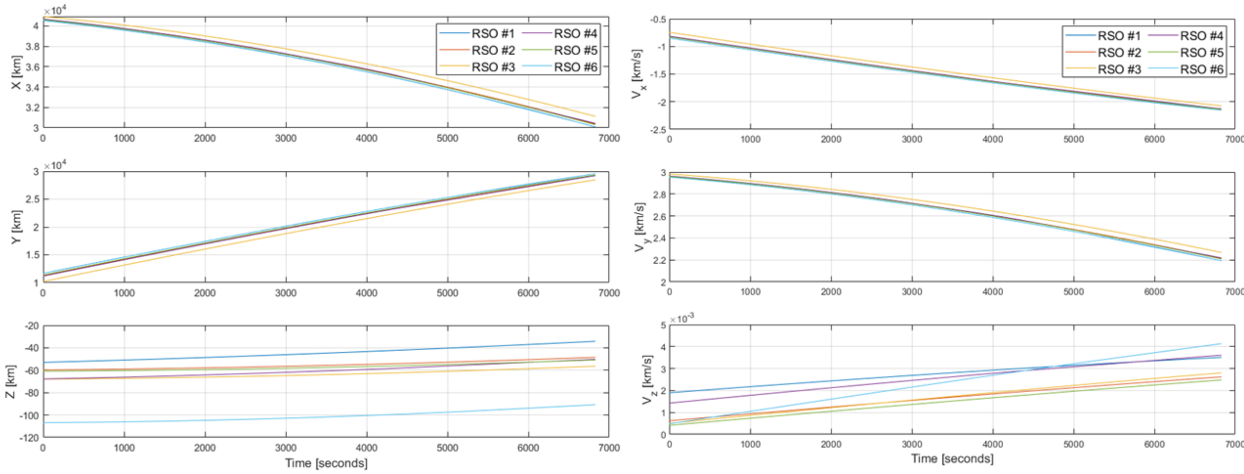

Fig. 16 UKF results position and velocity

The orbital elements at time of the final measurement were then calculated to compare with the corresponding orbital elements taken from TLE data. Note that all state vectors from TLEs must first be converted from the True Equator Mean Equinox (TEME) frame to the J2000 Earth Centered Inertial (ECI) frame before comparing orbital element values [22, 28]. Two Line Element data for each RSO detected was downloaded from space-track.org to provide the truth comparison [29]. Space-Track TLE data corresponding as close to the image epoch time as possible were used to minimize propagation errors for finding the "truth" comparison orbital elements. A MATLAB implementation of the Special General Perturbations 4 software was used to propagate the TLEs to the image epoch time and then state vectors were converted from the TEME frame to ECI state vectors at the appropriate times [30]. When converting from the TEME frame to ECI, rotations accounting for the Earth's nutation and precession must be accounted for, see Vallado chapter 3 for more details [22]. The epoch time of the final measurement was 2021-11-03 04:12:46.805 UTC. Orbital elements were calculated for all RSOs using the UKF mean states at the final time and compared to the orbital elements resulting from TLE data. The resulting difference in orbital elements is shown in Table 1. The NORAD ID's for each RSO are given in the first two columns for reference. Note that all angular differences are given in degrees. For the first four elements (a, e, i, Ω) the difference in the orbital elements is very small as these describe the shape and orientation of the orbit. Because all the RSOs are GEO objects in very near equatorial circular orbits, the comparison of the argument of perigee and true anomaly is not particularly helpful and thus these orbital elements are omitted from the comparison. The largest difference in semi-major axis is less than $0.13km$. The UKF estimated the size and shape of each orbit very well. Adding observations over a longer window should improve the estimation of the RAAN as well as the argument of perigee and true anomaly. Average processing time for each image was around 10 seconds. However,

Table 1 Orbital element difference from TLE Data

RSO #	NORAD ID	a $[km]$	e	i $[deg]$	Ω $[deg]$
1	42934	0.00952727	-1.32E-05	2.15E-06	-0.05188064
2	39008	0.001100883	-3.72E-06	9.90E-06	-0.083774096
3	43562	0.126258476	2.84E-05	2.26E-05	-0.084483232
4	36792	0.063955954	7.73E-06	-2.95E-07	-0.041614312
5	41592	0.015283389	-2.98E-06	-3.29E-07	-0.095538326
6	35942	0.00992726	6.66E-06	-2.15E-05	0.018696663

no attempt was made to optimize the MATLAB code to improve run times. Given an observational scenario where measurements are taken at this frequency real time operation could be achieved.

VII. Conclusion

Initial testing of the algorithms presented in this research show the viability of autonomous template image generation and RSO association across image frames. The ability to automatically update the template image when the template changes significantly, or matching failed was proven and provided excellent template matching results for the images tested. The algorithm was tested on a a large data set containing multiple RSOs with enough relative motion to require updating the template in order to successfully match across all frames. Initial orbit determination results show the capability of the algorithm to accept large amounts of data and process them to provide an orbit estimate only from the input of an image set (and associated image metadata such as image capture time). Further research to be done includes processing data sets that span longer times, as well as varying numbers of objects. Further orbit determination results are also the subject of future work including batch orbit estimation. Another area of future work is to utilize different methods of template matching and compare the results to normalized cross-correlation. Finally, software future work includes workflow optimization to use different hardware to improve processing speed and make the code more efficient.

Acknowledgments

This work was partially funded by the National Defense Science and Engineering Graduate fellowship program (NDSEG).

References

[1] Virtanen, J., Poikonen, J., Säntti, T., Komulainen, T., Torppa, J., Granvik, M., Muinonen, K., Pentikäinen, H., Martikainen, J., Näränen, J., Lehti, J., and Flohrer, T., "Streak detection and analysis pipeline for space-debris optical images," *Adv. Space Res.*, Vol. 57, No. 8, 2016, pp. 1607–1623.

[2] Schildknecht, T., "Optical surveys for space debris," *The Astronomy and Astrophysics Review*, Vol. 14, No. 1, 2007, pp. 41–111. Copyright - Springer-Verlag 2007; Last updated - 2014-08-22.

[3] Knutson, J., "Axios: U.S. says russian anti-satellite test created more than 1,500 piece of space debris." , 2021. URL `https://www.proquest.com/docview/2597529777?accountid=27203&forcedol=true`.

[4] Henderson, T. A., Mortari, D., and Davis, J., "Modifications to the Gooding Algorithm for Angles-only Initial Orbit Determination," *20TH AAS/AIAA Space Flight Mechanics Meeting*, 2010.

[5] Brad Sease, B. F., "GEODETICA: A general software platform for processing continuous space-based imagery," *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015.

[6] Guy Nir, Barak Zachay, and Eran O. Ofek, "Optimal and Efficient Streak Detection in Astronomical Images," *Astronomical Journal*, Vol. 156, No. 5, 2018, p. 229.

[7] Sease, B., and Flewelling, B., "Polar and spherical image transformations for star localization and RSO discrimination," *25th AAS/AIAA Space Flight Mechanics Meeting*, 2015.

[8] Lovell, T. A., Zuehlke, D., and Henderson, T., "Processing of Space Object Data from Optical Observers for Space Domain Awareness," *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–11. https://doi.org/10.1109/AERO50100.2021.9438157.

[9] Zuehlke, D., Henderson, T., Alan Lovell, T., and Sizemore, A., "An End-to-End Process for Local Space Situational Awareness from Optical Observers," *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1547–1555.

[10] Zuehlke, D., and Henderson, T., "Geostationary Satellite Constellation Tracking and Identification Using Normalized Cross Correlation," *AIAA Scitech 2020 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2020.

[11] Zuehlke, D. A., Henderson, T. A., and McMullen, S. A. H., "Machine learning using template matching applied to object tracking in video data," SPIE, 2019.

[12] Park, E.-S., Park, S.-Y., Roh, K.-M., and Choi, K.-H., "Satellite orbit determination using a batch filter based on the unscented transformation," *Aerosp. Sci. Technol.*, Vol. 14, No. 6, 2010, pp. 387–396.

[13] Sullivan, J., Lovell, T. A., and D'Amico, S., "Angles-only navigation for autonomous on-orbit space situational awareness applications," *AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT*, 2018.

[14] Tonc, L., and Richards, G., "Orbit Estimation from Angles-Only Observations Using Nonlinear Filtering Schemes," *Utah Space Grant Consortium*, 2019.

[15] Gooding, R. H., “A new procedure for orbit determination based on three lines of sight (angles only),” Tech. rep., Defence Research Agency Farnborough (United Kingdom), 1993.

[16] Gooding, R. H., “A new procedure for the solution of the classical problem of minimal orbit determination from three lines of sight,” *Celest. Mech. Dyn. Astron.*, Vol. 66, No. 4, 1996, pp. 387–423.

[17] Yang, Y., Chen, Z., Li, X., Guan, W., Zhong, D., and Xu, M., “Robust template matching with large angle localization,” *Neurocomputing*, Vol. 398, 2020, pp. 495–504.

[18] Kim, H. Y., and de Araújo, S. A., “Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast,” *Advances in Image and Video Technology*, Springer Berlin Heidelberg, 2007, pp. 100–113.

[19] Shi, J., and Tomasi, C., “Good features to track,” *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA*, 1994, pp. 593–600. https://doi.org/10.1109/CVPR.1994.323794.

[20] Lewis, J., “Fast Template Matching,” *Vis. Interface*, Vol. 95, 1994.

[21] “Normalized 2-D cross-correlation - MATLAB normxcorr2,” https://www.mathworks.com/help/images/ref/normxcorr2.html, 2021. Accessed: 2021-12-3.

[22] Vallado, D. A., and McClain, W. D., *Fundamentals of astrodynamics and applications*, Microcosm Press, 2013.

[23] Park, E.-S., Park, S.-Y., Roh, K.-M., and Choi, K.-H., “Satellite orbit determination using a batch filter based on the unscented transformation,” *Aerosp. Sci. Technol.*, Vol. 14, No. 6, 2010, pp. 387–396.

[24] Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., “A new approach for filtering nonlinear systems,” *Proceedings of 1995 American Control Conference - ACC’95*, Vol. 3, 1995, pp. 1628–1632 vol.3.

[25] Lang, D., Hogg, D. W., Mierle, K., Blanton, M., and Roweis, S., “Astrometry.net: Blind Astrometric Calibration Of Arbitrary Astronomical Images,” *The Astronomical Journal*, Vol. 139, No. 5, 2010, p. 1782–1800. https://doi.org/10.1088/0004-6256/139/5/1782.

[26] Zuehlke, D., “Space Image Processing and Orbit Estimation Using Small Aperture Optical Systems,” Ph.D. thesis, Embry-Riddle Aeronautical University, 2019.

[27] Richard Berry and the Celestron Engineering Team, “Big! Fast! Wide! Sharp! The Story of the Rowe-Ackermann Schmidt Astrograph,” Vol. 1, 2016.

[28] Seago, J., and Vallado, D., “Coordinate frames of the U.S. Space Object Catalogs,” *Astrodynamics Specialist Conference*, American Institute of Aeronautics and Astronautics, 2000, pp. 105–115.

[29] “Space-Track.Org,” , 2021. URL https://www.space-track.org.

[30] Vallado, David A, and Paul Crawford and Richard Hujsak and T.S. Kelso, “Revisiting Spacetrack Report #3: Rev 2,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006.