Advances in Resident Space Object Identification with Unresolved Optical Space Imagery

Abigail Miller[1] and Thomas Alan Lovell[2]
Embry-Riddle Aeronautical University, Daytona Beach, FL

Accurately identifying resident space objects within optical space imagery poses a key challenge in the realm of Space Situational Awareness. For unresolved imagery, the primary issue is distinguishing resident space objects from stars and other objects (or aberrations) that may be present. In this paper the authors improve an existing multi-step resident space object identification algorithm and enhance streamlined efforts such as faster runtime, greater versatility (applicability to a wide variety of imagery), and modularity (allowing the user to begin at any desired step in the process). Example results on real space imagery are presented.

I. Introduction

A fundamental tenet of Space Situational Awareness (SSA) is to properly identify resident space objects (RSOs) within unresolved optical space imagery. This subsequently allows the directional data, i.e. right ascension (RA) and declination (Dec), from the observer to each RSO to be processed resulting in an estimate of each RSO's orbit. The objective of this work is to enhance the developed streamlined algorithm from an existing framework to autonomously detect RSOs and determine their orbits, either onboard a spacecraft or from the ground.

To distinguish RSOs from the star field, a suggested practice is to perform multi-frame image collects and identify RSOs based on how the scene changes from frame to frame [1]. Two common modes of imaging are described as follows:

- Inertial stare mode: Here the camera's attitude is controlled to remain inertially fixed. Ideally, stars will remain stationary in the camera's field of view, but RSOs will move over time. For a long enough exposure time, the RSO will exhibit streaking in the image (see Figure 1).
- Orbit-following mode: Here the camera's attitude is controlled to follow an RSO whose orbit is known with at least some degree of accuracy. Ideally, the RSO will remain stationary in the camera's field of view, but stars will move over time (again, streaking for long enough exposure times) (see Figure 2).

In order to operate on these and other types of collects, an RSO identification (RSOID) algorithm must be robustly designed, firstly to distinguish actual objects in the images from false objects (i.e. collections of "noisy" pixels) and secondly, to distinguish RSOs from stars or other celestial objects.

Advances in the RSOID algorithm include investigations on specific streamlined RSOID steps and related functions. The streamlined RSOID algorithm supports a modular design by allowing the user to initiate the process from any specified step and fully complete the algorithm. The work presented explains each step defined in the streamlined algorithm and the modified approach implemented in the Plate Solving step.

[1] Master's Student, Physical Sciences Department, AIAA Student Member.
[2] Professor, Aerospace Engineering Department, AIAA Associate Fellow.

Figure 1. Unresolved image taken in internal state mode with RSO captures in the upper right.

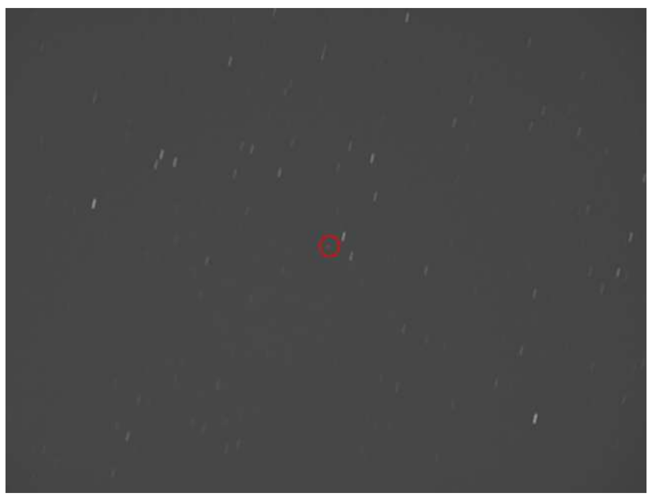

Figure 2. Unresolved image taken in orbit following mode with stationary RSO captured circled.

II. RSOID Methodology

The original version of RSOID [2-4] written in MATLAB was limited to the types of images and RSOs that could be processed, contained numerous extraneous functions, and required the user to perform a complete run of the algorithm every time regardless of whether a specific step in the algorithm was previously completed. The streamlined algorithm presented here consists of the following interdisciplinary steps:

1) Data Input collects the user's threshold preferences (described below), camera parameters, and the file location for the collect. The output is a structure containing the mentioned information as well as data-retrieving parameters for each step within the process.
2) Edge Detection is performed for each image in the collect, to determine the boundary between each "object" (i.e. set of illuminated pixels) and the blackness of space; thus, reducing each image to numerous segmented objects (actually point spread functions).
3) Centroiding is performed for each image in the collect, to localize each object in the camera frame based on pixel (x, y) coordinates.
4) Registration (or "Plate Solving") is performed for each image in the collect, to determine the inertial orientation of the camera and thereby convert the direction of each centroid in the camera frame (i.e. a unit vector represented by local pixel coordinates) into a direction expressed in terms of inertial coordinates (i.e., a unit vector represented by RA and Dec).
5) Star Elimination is performed on the collect, to remove all centroids identified as belonging to a star.

6) RSO Association is performed on the remaining centroids, to identify any RSOs in the collect and assign a set of centroids (i.e. a "tracklet") to each RSO.

All six steps explained are presented in the flowchart in Figure 3 below. The streamlined RSOID can be initialized at any of the six steps, allowing for modularity applications. The algorithm will automatically run the subsequent steps and save the output from every step to the user-directed path.

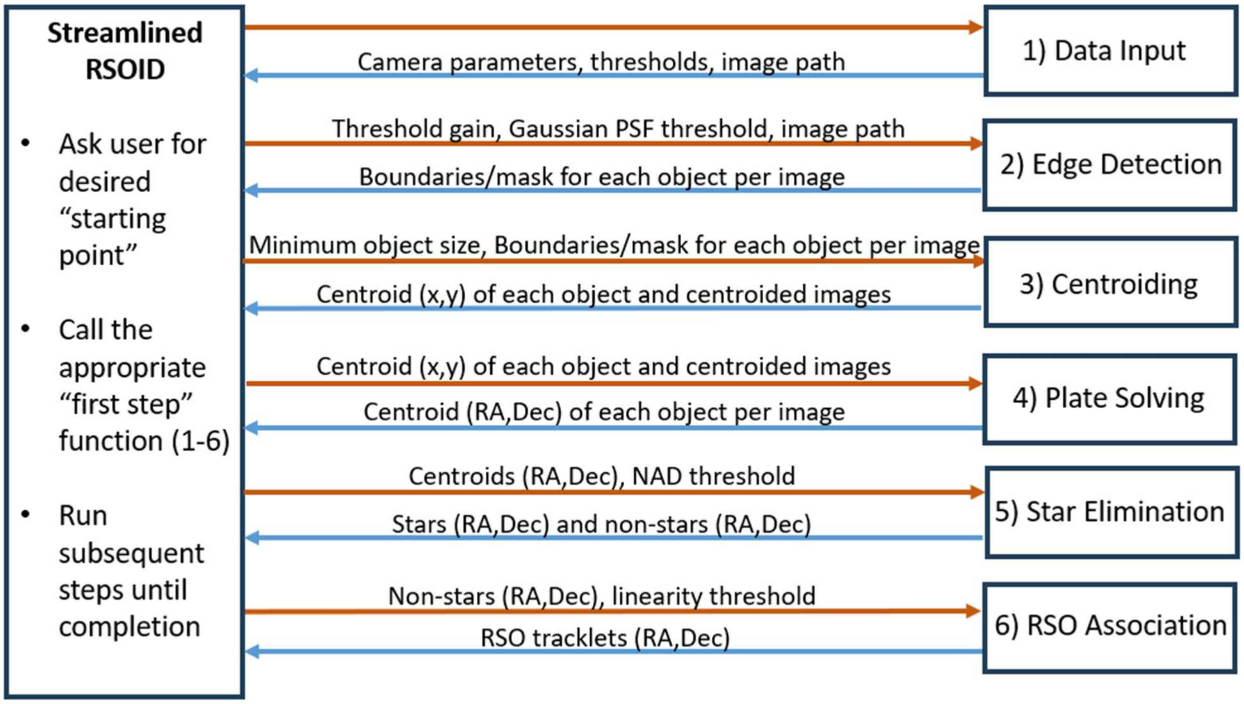

Figure 3. Flowchart of Streamlined RSOID structure. Orange arrows indicate the required input while blue indicates the output for each step.

III. Finalized Streamlined Steps

Implementing the streamlined RSOID process eliminates extraneous functions and supports a modular design. The current steps adapted to the streamlined paradigm are data input, edge detection, centroiding, and plate solving. The user can initiate the process at steps 1-4 with the algorithm automatically finishing the process. Previously, a user would be required to always start at the beginning of the process with no option to begin at various intermediate steps. The functions were layered with no separation and no modularity aspects with the original design. Data Input, Edge Detection, and Centroiding have been streamlined and presented previously [5]. Plate Solving is the most recent improvement to the streamlined format that follows modularity implications.

A. Data Input

Data Input is the first step of the streamlined process, which collects the images themselves, threshold values from user-defined parameters, and the location to save the output file for each step. This step consolidates all threshold and figure saving prompts to one prompt allowing for more efficient initial steps to begin the RSOID process. The user defined thresholds and their default values can be seen in Figure 4 below. The thresholds are defined for each step and the definition of these thresholds can be found [5].

Next, the user is prompted to upload their images and select the location for the output of each step to be exported. The final procedure for the Data Input step is the extraction of metadata from the images, namely the date and center of exposure time of each frame. The previous RSOID algorithm originally had six prompts for the user while the streamlined version has 3 initial prompts prior to running subsequent steps. Additionally, the previous algorithm was written to extract numerous other (extraneous) data types and was hard-coded to ingest certain mnemonics for each data type. The streamlined algorithm is much more robust in this regard. Once all of the data described above is collected, it is saved in a file named Data_Input.mat in the output file location originally chosen by the user

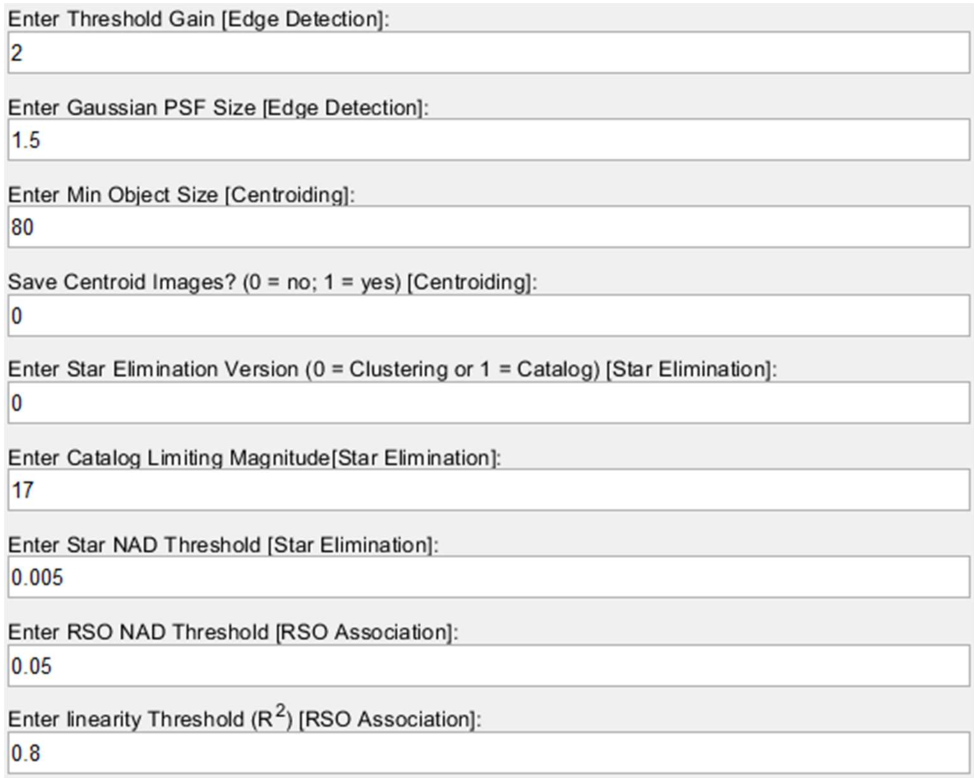
Enter Threshold Gain [Edge Detection]:
2
Enter Gaussian PSF Size [Edge Detection]:
1.5
Enter Min Object Size [Centroiding]:
80
Save Centroid Images? (0 = no; 1 = yes) [Centroiding]:
0
Enter Star Elimination Version (0 = Clustering or 1 = Catalog) [Star Elimination]:
0
Enter Catalog Limiting Magnitude[Star Elimination]:
17
Enter Star NAD Threshold [Star Elimination]:
0.005
Enter RSO NAD Threshold [RSO Association]:
0.05
Enter linearity Threshold (R^2) [RSO Association]:
0.8

Figure 4. User prompt displayed in Data Input step, applicable for all subsequent steps.

B. Edge Detection

Edge Detection is the second step for the streamlined RSOID process that determines boundaries around the illuminated objects in each image. When the user starts the process from the beginning (i.e. Step 1, Data Input), Edge Detection will automatically run after Data Input, but the streamlined version includes the option to begin the process at Edge Detection. That is, assuming the user during a prior run of RSOID ended the process after Step 1, a Data_Input.mat file would have been created and saved. When beginning the process at Edge Detection, the user is prompted to upload Data_Input.mat, which defines the values for each parameter in Figure 4, the path to the image files, and the path to the desired output files. At this point, the same dialog box as in Figure 4 is displayed, allowing the user to edit any parameters defined by Data_Input.mat. The algorithms process for Edge Detection is as follows for each image:

1) Application of intensity threshold: this sets all pixel intensity values below the noise floor to zero.
2) Image mask: matrix that defines the locations of pixels comprising each object (i.e. each set of contiguous illuminated pixels) detected from each image file.
3) Number of Detected Objects (NOBS): a vector containing the number of objects detected in each image.
4) Region location: confines all the locations for each detected object for each image.

The output structure for Edge Detection includes the vectors and matrices resulting from the above actions, as well as the values of each user-defined parameter or threshold (those displayed in Figure 4) at the time Edge Detection was executed. All of this output is saved in Edge_Detection_output.mat, which contains all the necessary data for the following step.

C. Centroiding

Centroiding takes the object information determined by Edge Detection and calculates the centroid of each detected object in each frame. When the user starts the process from the beginning, Centroiding will automatically run after Edge Detection, but the streamlined version includes the option to begin the process at Centroiding. In the latter case, the user is then prompted to upload Edge_Detection_output.mat, then the user can edit thresholds presented in Figure 4 from Centroiding through RSO Association. The number of threshold prompts decreases as the user moves forward through the streamlined RSOID process. Centroiding performs the following actions on the Edge Detection output:

1) Application of object size threshold: eliminates (i.e. ignores) each detected object whose "size" (i.e. number of contiguous illuminated pixels) is below the object size threshold.
2) Calculates the intensity-based centroid in pixel (*x, y*) coordinates of each detected object above the object size threshold.
3) Display and save centroid maps: If opted by the user, each image is displayed with indicators of all centroids detected by the algorithm overlaid on top of the image. These centroid map files are saved in the output location originally selected by the user.

The output structure for Centroiding includes the centroid locations of each detected object in each image and the values of each user-defined parameter or threshold at the time Centroiding was executed. All of this output is saved in Centroiding_output.mat, which contains all the necessary data for the following step.

D. Plate Solving

Plate Solving extracts the output of Centroiding and uses the open-source tool called Astrometry [6] to convert the pixel coordinates *(x, y)* for every centroid detected in each frame into RA and DEC values. The *(x, y)* pixel coordinates of a centroid represent a unit vector from the observer to the object in the camera's local coordinate frame, whereas the RA and Dec values represent the same unit vector converted into an inertial (J2000) frame. The algorithm will automatically perform this step after Centroiding, but the streamlined implication allows the user to begin at Plate Solving with the upload of Centroiding_output.mat. If the user starts at Plate Solving, the threshold prompt provides the user to edit thresholds from the Plate Solving step through RSO Association. Therefore, a smaller prompt than shown in Figure 4 will be displayed. Then the steps for Plate Solving to be complete are the following steps:

1) Creation of "quasi-image" files: for each frame in the collect, a randomly generated file name is given to represent a file in image format consisting of the (*x, y*) centroid locations solved in Step 3.
2) Conversion to RA and Dec: pass each quasi-image through Astrometry to calculate the RA and Dec coordinates of each centroid (i.e. object) in that image.
3) Extract values from two Astrometry output files within the MATLAB algorithm.

The streamlined Plate Solving technique calls Astrometry once for the solved output files that are temporarily saved in the current directory. Then Astrometry can interpret the *"wcs"* output file for a list of all RA and Dec values. Originally, Plate Solving would call out to Astrometry four times for every image input, causing a longer run time for the RA and Dec solutions. Streamlining this process included minimizing the call-outs to Astrometry and accessing all necessary data from two Astrometry output files.

To obtain the RA, Dec, and orientation angle of the center field of view (i.e. boresite) of the image, the *"wcs"* output file can be read through MATLAB to interpret the output file. The information we extract from the *"wcs"* file is the RA, Dec, and orientation matrix. From the orientation matrix, the orientation angle can be calculated. Utilizing these values, a Direction Cosine Matrix (DCM) is calculated.

For the list of RA and Dec values for every centroid object, the second call to Astrometry is required to interpret the *"wcs"* file. After this call, the *"full.rdls"* Astrometry file can be read through MATLAB to interpret the binary information within the file. This directly provides the RA and Dec values for every single centroid given to Astrometry from step three.

Once, all necessary information is extracted a figure of all RA and Dec values for each frame is created. This figure is presented within the Results section below. The final output structure for Plate Solving will consist of the input structure, DCM matrix, RA and Dec centroid list, and pointing vector (RA, Dec, θ) list for every frame. This gets compiled into a Plate_Solving_output.mat file prepared to execute the following step.

IV. Projected Progress

A. Star Elimination

Star Elimination takes the output of Plate Solving and determines which centroids in each frame are stars, so that the remaining centroids can go forward to the final step. This step of the process has recently been developed [7] to provide two options, known as clustering and catalog comparison. The former approach is based on the fact that for the duration of a collect, the RA and Dec from the observer to each star will remain essentially constant, while the RA and Dec from the observer to an RSO will change from one frame to the next. Therefore, the centroids across the frames of a collect representing the same star will tend to lie in a cluster closely spaced in RA and Dec, whereas this will not be true for any RSOs. The latter approach to Star Elimination involves direct comparison of each centroid location (in terms of RA and Dec) with that of known stars in a star catalog. If the user chooses the catalog comparison option, the authors currently utilize a portion of the Gaia catalog [8] that extends to 17^{th} magnitude. The authors have also experimented with the sparser Tycho-2 catalog [9], which extends to approximately 14th magnitude. The thresholds applicable to this step have been previously presented [5].

The plan is that when the user starts the process from the beginning, Star Elimination will automatically run after Plate Solving, but the option will exist to begin the process at Star Elimination. In the latter case, the user will be prompted to upload Plate_Solving_output.mat, after which the threshold prompt will display thresholds from Star Elimination through RSO Association for the user to edit (full threshold image shown as Figure 4). Star Elimination will perform the following action on the Plate Solving output:

1) Determination of star vs RSO centroids: for each frame in the collect, determine which centroids represent stars and which represent "non-stars" and parse them into two separate structures.

The output for Star Elimination includes the two structures mentioned above and the values of each user-defined parameter or threshold at the time Star Elimination was executed. This output is saved in Star_Elimination_output.mat, which contains all the necessary data for the following step.

B. RSO Association

For a typical collect, the centroids categorized by Star Elimination as "non-stars" should consist of any RSOs existing in the collect, along with any stars that may not have been successfully eliminated by the previous step, as well as groups of contiguous "noisy" pixels that have been incorrectly identified as objects by Edge Detection and Centroiding. The number of centroids comprising the latter two categories should not be very large, yet still likely greater than zero. Therefore, not all centroids will belong to RSOs. It is expected that for a collect whose duration is a few minutes or less, the path of the RSO in RA/Dec coordinates should be quasi-linear. Therefore, RSO Association assigns a set of centroids (i.e. a "tracklet") to each RSO by performing linear regression on each combination of non-stars across the frames of the collect.

The plan is that when the user starts the process from the beginning, RSO Association will automatically run after Star Elimination, but the option will exist to begin the process at RSO Association. In the latter case, the user will be prompted to upload Star_Elimination_output.mat, with a threshold prompt only showing the RSO Association thresholds for the user to edit (from Figure 4). Each combination of non-stars across the frames of the collect forms a path in RA and Dec. RSO Association will perform linear regression on the path, and if the coefficient of linearity (often referred to as "R^2") lies above $coeff_{min}$ (one of the thresholds in Figure 4), then this set of centroids is determined to belong to an RSO.

It should be noted that a particular RSO may not necessarily appear in every frame. For example, it may "blink" out during one or more frames if its reflectivity is varying over time (particularly true for a tumbling RSO). Alternately, the RSO may simply not be in the camera's field of view for the entire collect. Therefore, the RSO Association process does not require each tracklet to consist of a centroid from every frame.

V. Results from Real Image Collects

Unresolved space imagery has been obtained for several collects with an optical sensor from the authors' university. This consists of a Celestron RASA 11 inch telescope with an ASI 1600mm monochrome CMOS camera mounted on a Celestron CGE Pro mount. The field of view for this configuration is approximately 1.5° x 1.5°. In this section, RSOID results from a particular collect are presented. It is a 12-frame object-following collect spanning 23 seconds and following a specific object in the geostationary belt, AMAZONAS 2 (SatID 35942).

The first results presented to the user is the output from Centroiding. Figure 5 presents a centroid map obtained for the first frame of the collect, after the first three steps of RSOID (Data Input, Edge Detection, and Centroiding) have been executed. As alluded to above, the centroid map is a reproduction of the image itself, overlaid with green circles, with the center of each circle indicating the presence of a centroid. Note that most of the objects in the image exhibit a slight amount of streaking. This is quite common for an object-following collect, since a particular RSO is being followed during the exposure time of the image, therefore the stars do not appear stationary.

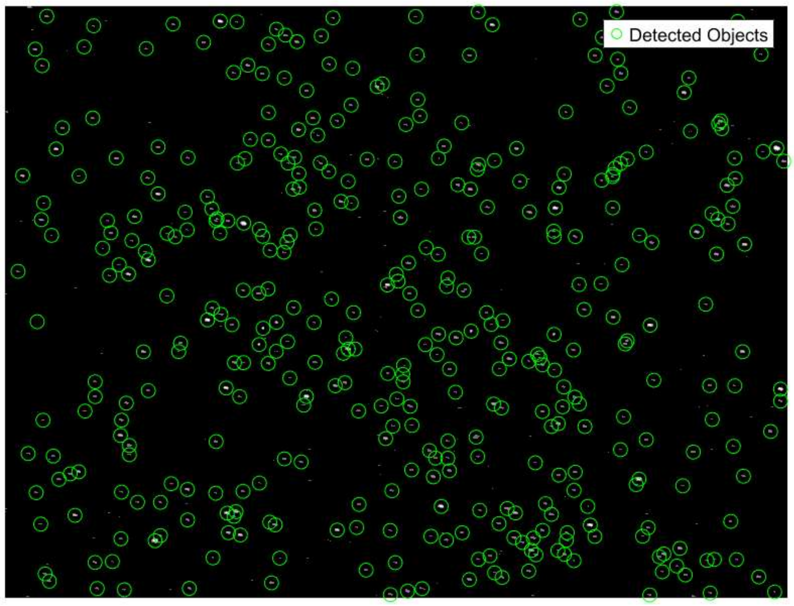

Figure 5. Centroid map obtained for the first frame of the collect.

As previously mentioned, once Plate Solving is completed, a figure of all RA and Dec values is shown as Figure 6 below. Essentially, this figure is a RA and Dec map plotting all centroids from each frame into a single image. Using the jet color scheme in MATLAB, the centroids in Frame 1 are plotted as blue circles, the centroids in the last frame are plotted as red circles, and the shades of color used to plot all frames in between are linearly spaced across the frequency spectrum. Therefore, each star in the image tends to be represented by a cluster of closely spaced centroids, with the red circle at the top, since the last frame of the collect is last set of points overlaid on the plot. Whereas each RSO tracklet is represented by a set of circles forming a blue-to-red quasi-linear path.

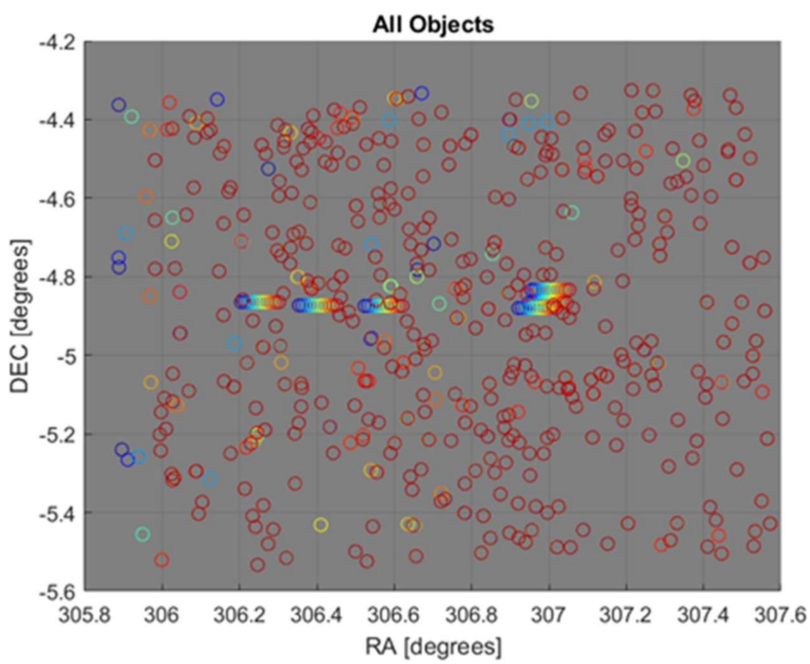

Figure 6. Aggregation of all centroids from each frame of the collect superimposed together.

After Star Elimination is executed (with the "clustering" option chosen), Figure 7 displays all centroids determined to be non-stars. As expected, the majority of non-stars are seen to belong to an RSO. Finally, after RSO association is executed, Figure 8 displays only the centroids belonging to each of the 6 RSOs contained in the collect, with a different color assigned to each RSO.

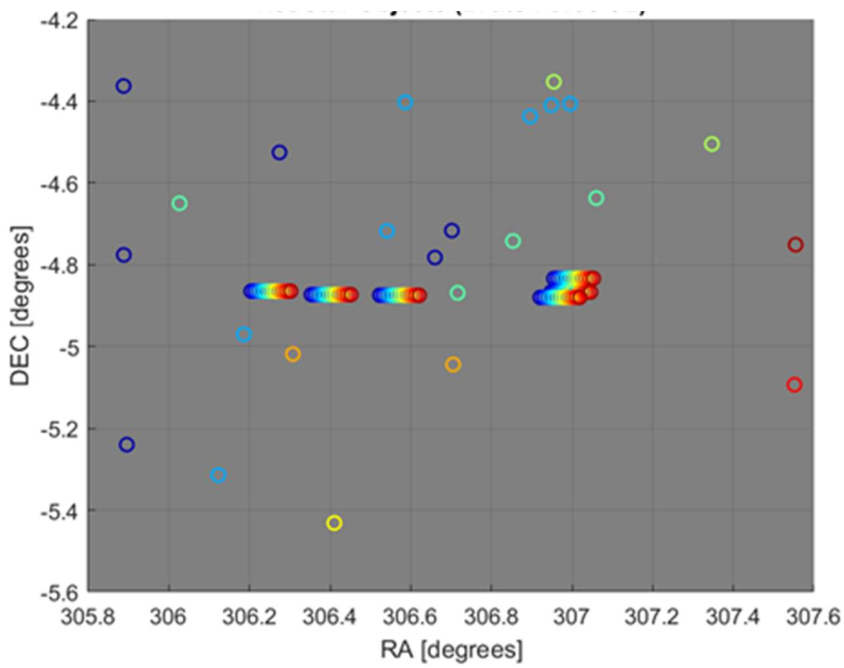

Figure 7. Centroids remaining in the collect after Star Elimination based on clustering.

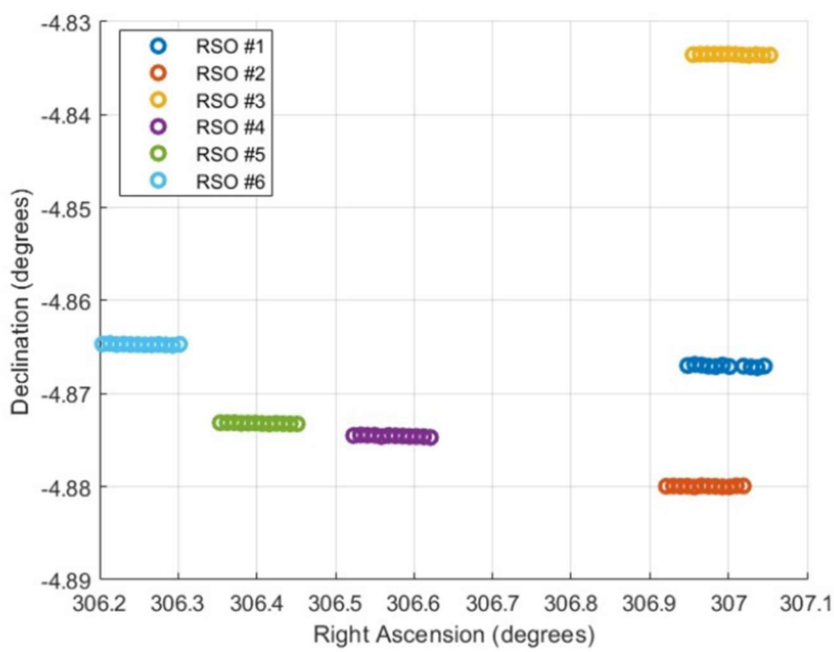

Figure 8. Correctly identified RSOs in the collect after RSO Association.

VI. Modularity Strategy

As mentioned throughout the paper, a major contribution of this work is the modularity of the RSOID algorithm, i.e. the ability for the user to begin at any step of the process. There are two major advantages to this, the first being from a research perspective. This adaptive design allows the user to investigate later steps without starting at the beginning of the script every time. This reduces the runtime because the user can edit thresholds prior to any specific step that is initiated and can begin directly with research investigations.

The second advantage of the modular scheme comes from an applied (i.e. mission) perspective. Due to the compact nature of the algorithm, RSOID could become a major contributor to onboard autonomous SSA capability, whereby all or part of the RSOID process could be performed on the imagery obtained onboard the spacecraft itself, rather than being performed on the ground. The amount of autonomy is up to the user's discretion and allows the mission team to draw the line between onboard and ground operations wherever they want, and the chosen paradigm can be altered at any time by uplinking an update to the flight software. Clearly, the higher the level of autonomy the spacecraft possesses, the less data required to be downlinked, as each succeeding step of the RSOID process entails a progressively smaller data set.

It should also be noted that the modular nature of the algorithm allows efficient troubleshooting when the results of a collect appear suspect. For example, if the final result indicates a much higher or lower number of RSOs in the collect than was expected, or the path of one or more RSOs is erratic, etc, the algorithm allows the user to trace back and pinpoint where and why this occurred by examining thoroughly the output of each step. A further explanation of the modularity aspect can be found in [5].

VII. Conclusion and Future Work

This paper has detailed the streamlining effort underway for a resident space object identification algorithm that operates on unresolved space imagery. The goal is for the algorithm to be written cleanly and in modular fashion. Each step of the algorithm has been clearly detailed, as well as an explanation of how the algorithm accommodates the user being able to begin at any step of the process, assuming output for the previous steps of the process has already been obtained. Specifically, significant improvements to the Plate Solving step of the process have been explained and how they meet the modularity design. The obvious future task is to finish streamlining the algorithm and perhaps refine one or more steps of the process by a moderate amount. The authors possess ample image collects on which to test extensively the full streamlined algorithm. From a research perspective, if the user desires to run the algorithm multiple times on a particular collect as part of an experiment, an additional improvement to the code would be to allow the user to upload an existing Plate Solving solution for each frame of the collect and essentially remove Plate Solving from the sequence of steps. The logic here is that regardless what values are chosen for thresholds and other user defined parameters, the orientation of the camera for each frame will not change. Overall, the algorithm shows significant promise as a candidate for onboard Space Situational Awareness operations.

References

[1] Zuehlke, D., "Space Image Processing and Orbit Estimation Using Small Aperture Optical Systems," Embry-Riddle Aeronautical University, PhD Dissertation, Apr. 2023.

[2] 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), IEEE, Mar. 2021, pp. 1–11, 10.1109/AERO50100.2021.9438157.

[3] Zuehlke, D., Henderson, T., Lovell, T.A., 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), Apr. 2020, pp. 1547–1555, 10.1109/PLANS46316.2020.9109868.

[4] Anderson, J., Anderson, A., Zuehlke, D., Canales, D., and Lovell, T.A., "Resident Space Object Identification in Arbitrary Space Images," Paper AAS 23-179, AAS/AIAA Spaceflight Mechanics Meeting, Austin TX, 2023.

[5] Miller, A., Lovell, A., "A Refined Approach to Resident Space Object Identification in Unresolved Optical Space Imagery," AIAA 2026-0204. AIAA SCITECH 2026 Forum. January 2026.

[6] Lang , D., Hogg, D. W., Mierle, K., Blanton, M., and Roweis, S., "Astrometry.net: Blind astrometric calibration of arbitrary astronomical images," The Astronomical Journal 139, pp 1782–1800, Oct. 2009.

[7] Pavetto-Stewart, E. and Lovell, T.A., "Advancing Star Elimination Methods for Improved Resident Space Object Detection in Space Situational Awareness," Paper AAS 25-556, AAS Astrodynamics Specialist Conference, Boston MA, 2025.

[8] Gaia ESA Archive, https://gea.esac.esa.int/archive/ [retrieved Feb 2025].

[9] Hog E., Fabricius C., Makarov V.V., Urban S., Corbin T., Wycoff G., Bastian U., Schwekendiek P., Wicenec A., "The Tycho-2 Catalogue of the 2.5 Million Brightest Stars," Astronomy and Astrophysics, v.355, p.L27-L30, March 2000.